



**TUGAS AKHIR - KI151601**

# **IMPLEMENTASI EKSTRAKSI FITUR UNTUK PENGELOMPOKAN BERKAS MUSIK BERDASARKAN KEMIRIPAN KARAKTERISTIK SUARA**

**RAMADHAN ROSIHADI PERDANA  
NRP 5112100032**

**Dosen Pembimbing I  
Dr.Eng. Chastine Fatichah, S.Kom, M.Kom.**

**Dosen Pembimbing II  
Rully Soelaiman, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**





**TUGAS AKHIR - KI151601**

**IMPLEMENTASI EKSTRAKSI FITUR UNTUK  
PENGELOMPOKAN BERKAS MUSIK BERDASARKAN  
KEMIRIPAN KARAKTERISTIK SUARA**

**RAMADHAN ROSIHADI PERDANA  
NRP 5112100032**

**Dosen Pembimbing I  
Dr.Eng. Chastine Fatichah, S.Kom, M.Kom.**

**Dosen Pembimbing II  
Rully Soelaiman, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017**

*[Halaman ini sengaja dikosongkan]*



**UNDERGRADUATE THESES - KI151601**

# **IMPLEMENTATION OF FEATURES EXTRACTION FOR MUSIC FILES CLASSIFICATION BASED ON SIMILAR CHARACTERISTIC OF SOUND**

**RAMADHAN ROSIHADI PERDANA  
NRP 5112100032**

**Supervisor I  
Dr.Eng. Chastine Fatichah, S.Kom, M.Kom.**

**Supervisor II  
Rully Soelaiman, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2017**

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

### IMPLEMENTASI EKSTRAKSI FITUR UNTUK PENGELOMPOKAN BERKAS MUSIK BERDASARKAN KEMIRIPAN KARAKTERISTIK SUARA

#### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Komputasi Cerdas dan Visualisasi  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**RAMADHAN ROSIHADI PERDANA**

NRP : 5112 100 032

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

NIP: 197107182006041001

(Pembimbing 1)

Rully Soelaiman, S.Kom., M.Kom.

NIP: 197002131994021001

(Pembimbing 2)

**SURABAYA  
JANUARI 2017**

*[Halaman ini sengaja dikosongkan]*



# IMPLEMENTASI EKSTRAKSI FITUR UNTUK PENGELOMPOKAN BERKAS MUSIK BERDASARKAN KEMIRIPAN KARAKTERISTIK SUARA

Nama Mahasiswa : RAMADHAN ROSIHADI PERDANA  
NRP : 5112100032  
Jurusan : Teknik Informatika FTIF-ITS  
Dosen Pembimbing 1 : Dr.Eng. Chastine Fatichah, S.Kom,  
M.Kom.  
Dosen Pembimbing 2 : Rully Soelaiman, S.Kom., M.Kom.

## *Abstrak*

*Pengelompokkan musik berdasarkan karakteristik suara merupakan hal penting bagi penikmat musik.. Penikmat musik tidaklah mencari musik berdasarkan artis tetapi juga mencari musik berdasarkan genre yang diinginkannya. Karena itu dibutuhkan metode pengelompokkan yang tepat untuk mengkategorikan musik berdasarkan genre secara otomatis.*

*Tugas akhir ini melakukan pengelompokkan musik berdasarkan genre. Dengan mengekstraksi fitur spectral centroid, spectral flux, spectral rolloff, dan short time energy pada tiap berkas musik yang diolah dan kemudian dihitung nilai mean, median, skewness, dan kurtosisnya. Dan selanjutnya dikelompokkan menggunakan metode klasifikasi Random Forest dengan alat bantu Weka.*

*Uji coba dilakukan dengan menggunakan kombinasi nilai atribut komponen ekstraksi fitur dan berkas musik yang berbeda-beda sesuai genre. Hasil uji coba klasifikasi pada Tugas Akhir ini menghasilkan nilai akurasi terbaik sebesar 80,47%.*

**Kata kunci:** *Ekstraksi Fitur Audio, Musik, Klasifikasi.*

*[Halaman ini sengaja dikosongkan]*

# IMPLEMENTATION OF FEATURES EXTRACTION FOR MUSIC FILES CLASSIFICATION BASED ON SIMILAR CHARACTERISTIC OF SOUND

**Student's Name** : RAMADHAN ROSIHADI PERDANA  
**Student's ID** : 5112100032  
**Department** : Teknik Informatika FTIF-ITS  
**First Advisor** : Dr.Eng. Chastine Fatichah, S.Kom,  
M.Kom.  
**Second Advisor** : Rully Soelaiman, S.Kom., M.Kom.

## *Abstract*

*Music classification based on voice characteristics is essential for music lovers. Connoisseurs of music is not looking for music just by artists but also search for music by genre he/she wants. Therefore we need a method of grouping the right to categorize music by genre automatically.*

*This final task is grouping music by genre. By extracting feature spectral centroid, spectral flux, spectral rolloff, spectral flatness, zero crossing, and short-time energy. Of each music file is processed and then calculated the mean, median, skewness, and kurtosis. And further classified using Random Forest classification methods with Weka .*

*The test is done using a combination of the component of features extraction process and music files based on genre. The classification accuracy of 80,47% is achieved in this work..*

**Keywords:** *Audio Features Extraction, Music, Classification*

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji dan syukur bagi Allah SWT, yang telah melimpahkan rahmat, dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“IMPLEMENTASI EKSTRAKSI FITUR UNTUK PENGELOMPOKAN BERKAS MUSIK BERDASARKAN KEMIRIPAN KARAKTERISTIK SUARA”**.

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang berharga bagi penulis. Dengan pengerjaan Tugas Akhir, penulis dapat memperdalam, meningkatkan, serta menerapkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS.

Terselesaikannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan dari berbagai pihak. Dan dalam kesempatan ini penulis mengucapkan rasa syukur dan terima kasih kepada:

1. Allah SWT, karena atas izin-Nya lah penulis dapat menyelesaikan Tugas Akhir dengan baik.
2. Nabi Muhammad SAW, yang telah memberikan banyak sekali teladan dan inspirasi bagi penulis.
3. Papa, mama, dan adik penulis yang senantiasa memberikan banyak semangat, dukungan, doa, dan perhatian kepada penulis. Buku ini penulis persembahkan secara khusus untuk Papa, Mama, dan Adik.
4. Bapak Rully Soelaiman, S.Kom., M.Kom. selaku pembimbing II Tugas Akhir penulis yang telah membimbing, membantu, dan memotivasi penulis sehingga dapat menyelesaikan Tugas Akhir. Terima kasih atas segala ilmu dan nasihat yang bapak berikan kepada penulis sehingga menjadikan penulis menjadi individu yang lebih baik dan matang.

5. Ibu Dr.Eng. Chastine Fatichah, S.Kom, M.Kom. selaku pembimbing I Tugas Akhir penulis yang telah membimbing, membantu, dan memotivasi penulis sehingga dapat menyelesaikan Tugas Akhir.
6. Qonita Luthfia Sutino, yang telah berbaik hati memberi dukungan dan motivasi kepada penulis selama mengerjakan Tugas Akhir ini.
7. Bapak Radityo Anggoro, S.Kom., M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah membagikan ilmu kepada penulis.
8. Teman-teman angkatan 2012 Jurusan Teknik Informatika ITS, yang telah menjadi teman seperjuangan dalam suka dan duka selama menjalani kuliah.

Surabaya, Januari 2017

Ramadhan Rosihadi Perdana

## DAFTAR ISI

|  |      |
|--|------|
| LEMBAR PENGESAHAN... <b>Error! Bookmark not defined.</b> |      |
| Abstrak .....  | vii  |
| Abstract .....   | viii |
| KATA PENGANTAR.....                                      | x    |
| DAFTAR ISI.....  | xii  |
| DAFTAR GAMBAR .....                                      | xvi  |
| DAFTAR TABEL .....                                       | xxii |
| DAFTAR KODE SUMBER.....                                  | xxiv |
| BAB I PENDAHULUAN .....                                  | 1    |
| 1.1 Latar Belakang.....                                  | 1    |
| 1.2 Rumusan Masalah.....                                 | 2    |
| 1.3 Batasan Masalah .....                                | 2    |
| 1.4 Tujuan2 .....  |      |
| 1.5 Manfaat.....   | 3    |
| 1.6 Metodologi .....                                     | 3    |
| 1.7 Sistematika Penulisan Laporan Tugas Akhir .....      | 4    |
| BAB II TINJAUAN PUSTAKA .....                            | 7    |
| 2.1 Berkas MP3 .....                                     | 7    |
| 2.2 Ekstraksi fitur .....                                | 8    |
| 2.2.1 Spectral Centroid .....                            | 9    |
| 2.2.2 Spectral Flux .....                                | 10   |
| 2.2.3 Spectral Rolloff .....                             | 10   |
| 2.2.4 Short Time Energy .....                            | 11   |
| 2.2.5 Zero Crossing .....                                | 12   |
| 2.2.6 Spectral Flatness.....                             | 12   |
| 2.2.7 Discrete Fourier Transformation (DFT) .....        | 13   |
| 2.2.8 Fast Fourier Transform (FFT).....                  | 13   |
| 2.2.9 Hamming Window .....                               | 15   |
| 2.2.10 Mean.....   | 16   |
| 2.2.11 Deviasi Standar.....                              | 17   |
| 2.2.12 Skewness .....                                    | 17   |
| 2.2.13 Kurtosis .....                                    | 18   |
| 2.3 Klasifikasi Random Forest.....                       | 20   |

|   |            |
|---|------------|
| <b>BAB III PERANCANGAN .....</b>                                  | <b>23</b>  |
| 3.1 Pra Proses .....  | 24         |
| 3.2 Ekstraksi Fitur .....   | 26         |
| 3.2.1 Program Utama Ekstraksi Fitur .....                         | 30         |
| 3.2.2 Spectral Centroid.....                                      | 33         |
| 3.2.3 Spectral Flux .....   | 34         |
| 3.2.4 Spectral Rolloff .....                                      | 36         |
| 3.2.5 Short Time Energy .....                                     | 37         |
| 3.2.6 Zero Crossing .....   | 39         |
| 3.2.7 Spectral Flatness.....                                      | 40         |
| 3.3 Perancangan Proses Klasifikasi .....                          | 42         |
| <b>BAB IV IMPLEMENTASI.....</b>                                   | <b>45</b>  |
| 4.1 Lingkungan Implementasi .....                                 | 45         |
| 4.2 Implementasi .....  | 45         |
| 4.2.1 Implementasi Program Utama Tahapan Pra<br>Proses .....      | 45         |
| 4.2.2 Implementasi Program Utama Tahapan<br>Ekstraksi Fitur ..... | 46         |
| 4.2.3 Ekstraksi Fitur Spectral Flatness .....                     | 51         |
| 4.2.4 Implementasi Klasifikasi .....                              | 52         |
| <b>BAB V UJI COBA DAN EVALUASI.....</b>                           | <b>53</b>  |
| 5.1 Lingkungan Pelaksanaan Uji Coba .....                         | 53         |
| 5.2 Data Uji Coba .....   | 53         |
| 5.3 Uji Coba Ekstraksi Fitur .....                                | 54         |
| 5.3.1 Uji Coba Ekstraksi Fitur Spectral Flux .....                | 55         |
| 5.3.2 Uji Coba Ekstraksi Fitur Spectral Centroid .....            | 61         |
| 5.3.3 Uji Coba Ekstraksi Fitur Zero Crossing.....                 | 68         |
| 5.3.4 Uji Coba Ekstraksi Fitur Spectral Rolloff....               | 74         |
| 5.3.5 Uji Coba Ekstraksi Fitur Short Time<br>Energy .....         | 81         |
| 5.3.6 Uji Coba Ekstraksi Fitur Spectral Flatness ..               | 87         |
| 5.4 Skenario dan Evaluasi Pengujian.....                          | 93         |
| 5.4.1 Uji Coba dan Evaluasi Skenario 1 .....                      | 94         |
| 5.4.2 Uji Coba dan Evaluasi Skenario 2 .....                      | 96         |
| 5.4.3 Uji Coba dan Evaluasi Skenario 3 .....                      | 98         |
| <b>BAB VI KESIMPULAN DAN SARAN .....</b>                          | <b>101</b> |

|                      |     |
|----------------------|-----|
| 6.1. Kesimpulan..... | 101 |
| 6.2. Saran .....     | 101 |
| DAFTAR PUSTAKA ..... | 103 |



*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.2.1 Sinyal suara digital pada berkas mp3 lagu berjudul “Stormy Blues” oleh Arne Bang Huseby .....                            | 14 |
| Gambar 2.2.2 Hasil transformasi fourier sinyal suara digital pada berkas mp3 lagu berjudul “Stormy Blues” oleh Arne Bang Huseby ..... | 15 |
| Gambar 2.2.3 Spektrum dengan hamming window .....   | 16 |
| Gambar 2.2.4 Perbedaan grafik kurva berdasarkan nilai skewness. ....  | 18 |
| Gambar 2.2.5 Perbedaan kategori gravik kurva berdasarkan nilai kurtosis .....   | 19 |
| Gambar 2.3.1 Proses klasifikasi Random Forest. ....   | 21 |
| Gambar 2.3.1 Diagram alir keseluruhan proses .....  | 24 |
| Gambar 3.1.1 Diagram alir tahapan pra proses .....  | 25 |
| Gambar 3.2.1 Diagram alir tahap ekstraksi fitur .....   | 27 |
| Gambar 3.2.2 Pseudocode program utama (bagian pertama) ..   | 31 |
| Gambar 3.2.3 Pseudocode program utama (bagian kedua) ...  | 32 |
| Gambar 3.2.4 Diagram alir proses ekstraksi fitur spectral centroid .....  | 33 |
| Gambar 3.2.5 Pseudocode program ekstraksi fitur spectral centroid .....   | 34 |
| Gambar 3.2.6 Pseudocode program ekstraksi fitur spectral flux (bagian pertama) .....  | 34 |
| Gambar 3.2.7 Pseudocode program ekstraksi fitur spectral flux (bagian kedua) .....  | 35 |
| Gambar 3.2.8 Diagram alir proses spectral flux .....  | 35 |
| Gambar 3.2.9 Diagram alur proses ekstraksi fitur spectral rolloff .....   | 36 |
| Gambar 3.2.10 Pseudocode program ekstraksi fitur spectral rolloff .....   | 37 |
| Gambar 3.2.11 Pseudocode program ekstraksi fitur short time energy .....  | 37 |

|   |    |
|---|----|
| Gambar 3.2.12 Diagram alir proses ekstraksi fitur short time energy .....                           | 38 |
| Gambar 3.2.13 Diagram alir proses ekstraksi fitur zero crossing .....                               | 39 |
| Gambar 3.2.14 Pseudocode program ekstraksi fitur zero crossing .....                                | 40 |
| Gambar 3.2.15 Pseudocode program ekstraksi fitur spectral flatness (bagian pertama) .....           | 40 |
| Gambar 3.2.16 Pseudocode program ekstraksi fitur spectral flatness (bagian kedua) .....             | 41 |
| Gambar 3.2.17 Diagram alir proses ekstraksi fitur spectral flatness .....                           | 41 |
| Gambar 5.3.1 Histogram nilai fitur spectral flux pada berkas mp3 buku audio .....                   | 55 |
| Gambar 5.3.2 Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre metal .....     | 56 |
| Gambar 5.3.3 Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre blues .....     | 56 |
| Gambar 5.3.4 Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre classical ..... | 57 |
| Gambar 5.3.5 Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre country .....   | 57 |
| Gambar 5.3.6 Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre disco .....     | 58 |
| Gambar 5.3.7 Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre hip hop .....   | 58 |
| Gambar 5.3.8 Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre jazz .....      | 59 |
| Gambar 5.3.9 Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre pop .....       | 59 |
| Gambar 5.3.10 Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre reggae .....   | 60 |
| Gambar 5.3.11 Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre rock .....     | 60 |
| Gambar 5.3.12 Histogram nilai fitur spectral centroid pada berkas mp3 buku audio .....              | 62 |

|  |    |
|--|----|
| Gambar 5.3.13 Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre metal.....      | 62 |
| Gambar 5.3.14 Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre blues .....     | 63 |
| Gambar 5.3.15 Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre classical ..... | 63 |
| Gambar 5.3.16 Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre country .....   | 64 |
| Gambar 5.3.17 Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre disco .....     | 64 |
| Gambar 5.3.18 Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre hip hop.....    | 65 |
| Gambar 5.3.19 Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre jazz .....      | 65 |
| Gambar 5.3.20 Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre pop.....        | 66 |
| Gambar 5.3.21 Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre reggae .....    | 66 |
| Gambar 5.3.22 Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre rock .....      | 67 |
| Gambar 5.3.23 Histogram nilai fitur zero crossing pada berkas mp3 buku audio.....                        | 68 |
| Gambar 5.3.24 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre metal .....         | 69 |
| Gambar 5.3.25 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre blues .....         | 69 |
| Gambar 5.3.26 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre classical .....     | 70 |
| Gambar 5.3.27 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre country.....        | 70 |
| Gambar 5.3.28 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre disco .....         | 71 |
| Gambar 5.3.29 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre hip hop.....        | 71 |
| Gambar 5.3.30 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre jazz .....          | 72 |

|  |    |
|--|----|
| Gambar 5.3.31 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre pop.....            | 72 |
| Gambar 5.3.32 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre reggae .....        | 73 |
| Gambar 5.3.33 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre rock.....           | 73 |
| Gambar 5.3.34 Histogram nilai fitur zero crossing pada berkas mp3 buku audio.....                        | 75 |
| Gambar 5.3.35 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre metal.....          | 75 |
| Gambar 5.3.36 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre blues .....         | 76 |
| Gambar 5.3.37 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre classical .....     | 76 |
| Gambar 5.3.38 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre country.....        | 77 |
| Gambar 5.3.39 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre disco .....         | 77 |
| Gambar 5.3.40 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre hip hop.....        | 78 |
| Gambar 5.3.41 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre jazz .....          | 78 |
| Gambar 5.3.42 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre pop.....            | 79 |
| Gambar 5.3.43 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre reggae .....        | 79 |
| Gambar 5.3.44 Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre rock.....           | 80 |
| Gambar 5.3.45 Histogram nilai fitur short time energy pada berkas mp3 buku audio .....                   | 81 |
| Gambar 5.3.46 Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre metal.....      | 82 |
| Gambar 5.3.47 Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre blues .....     | 82 |
| Gambar 5.3.48 Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre classical ..... | 83 |

|  |    |
|--|----|
| Gambar 5.3.49 Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre country .....   | 83 |
| Gambar 5.3.50 Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre disco .....     | 84 |
| Gambar 5.3.51 Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre hip hop.....    | 84 |
| Gambar 5.3.52 Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre jazz .....      | 85 |
| Gambar 5.3.53 Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre pop.....        | 85 |
| Gambar 5.3.54 Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre reggae .....    | 86 |
| Gambar 5.3.55 Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre rock .....      | 86 |
| Gambar 5.3.56 Histogram nilai fitur short time energy pada berkas mp3 buku audio .....                   | 88 |
| Gambar 5.3.57 Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre metal.....      | 88 |
| Gambar 5.3.58 Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre blues .....     | 89 |
| Gambar 5.3.59 Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre classical ..... | 89 |
| Gambar 5.3.60 Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre country .....   | 90 |
| Gambar 5.3.61 Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre disco .....     | 90 |
| Gambar 5.3.62 Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre hip hop.....    | 91 |
| Gambar 5.3.63 Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre jazz .....      | 91 |
| Gambar 5.3.64 Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre pop.....        | 92 |
| Gambar 5.3.65 Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre reggae .....    | 92 |
| Gambar 5.3.66 Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre rock .....      | 93 |

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 3.2.1 Daftar Variabel yang Digunakan Pada Pseudocode Ekstraksi Fitur (Bagian Pertama) .....   | 28 |
| Tabel 3.2.2 Daftar Variabel yang Digunakan Pada Pseudocode Ekstraksi Fitur (Bagian Kedua) .....   | 29 |
| Tabel 3.2.3 Daftar Fungsi yang Digunakan Pada Pseudocode program Ekstraksi Fitur (Bagian Pertama) .....                                   | 29 |
| Tabel 3.2.4 Daftar Fungsi yang Digunakan Pada Pseudocode program Ekstraksi Fitur (Bagian Kedua) .....                                     | 30 |
| Tabel 3.3.1 Daftar parameter pada alat bantu Weka untuk menjalankan proses klasifikasi dengan metode Random Forest (bagian pertama) ..... | 42 |
| Tabel 3.3.2 Daftar parameter pada alat bantu Weka untuk menjalankan proses klasifikasi dengan metode Random Forest (bagian kedua) .....   | 43 |
| Tabel 5.1.1 Lingkungan Uji Coba .....   | 53 |
| Tabel 5.3.1 Hasil perhitungan nilai statistik data nilai fitur spectral flux pada frame-frame yang terbentuk .....                        | 61 |
| Tabel 5.3.2 Hasil perhitungan nilai statistik data nilai fitur spectral centroid pada frame-frame yang terbentuk .....                    | 67 |
| Tabel 5.3.3 Hasil perhitungan nilai statistik data nilai fitur zero crossing pada frame-frame yang terbentuk .....                        | 74 |
| Tabel 5.3.4 Hasil perhitungan nilai statistik data nilai fitur spectral rolloff pada frame-frame yang terbentuk .....                     | 80 |
| Tabel 5.3.5 Hasil perhitungan nilai statistik data nilai fitur short time energy pada frame-frame yang terbentuk .....                    | 87 |
| Tabel 5.4.1 Confusion matrix hasil klasifikasi Random Forest dengan nilai step saat ekstraksi fitur sebesar 0.5 .....                     | 95 |
| Tabel 5.4.2 Confusion matrix untuk hasil klasifikasi Random Forest dengan nilai step saat ekstraksi fitur sebesar 1 .....                 | 95 |
| Tabel 5.4.3 Nilai akurasi hasil proses klasifikasi Random Forest uji coba skenario 1 .....  | 96 |
| Tabel 5.4.4 Confusion matrix untuk hasil uji coba skenario ketiga dengan menggunakan data Kombinasi Genre 1 .....                         | 97 |



|  |    |
|--|----|
| Tabel 5.4.5 Confusion matrix untuk hasil uji coba skenario ketiga dengan menggunakan data Kombinasi Genre 2 (Bagiam pertama) ..... | 97 |
| Tabel 5.4.6 Confusion matrix untuk hasil uji coba skenario ketiga dengan menggunakan data Kombinasi Genre 2 (Bagiam kedua) .....   | 98 |
| Tabel 5.4.7 Nilai akurasi hasil proses klasifikasi Random Forest uji coba skenario 2.....  | 98 |
| Tabel 5.4.8 Perbandingan nilai akurasi berdasarkan metode klasifikasi yang digunakan pada skenario 3 .....                         | 99 |

## DAFTAR KODE SUMBER

|  |    |
|--|----|
| Kode Sumber 4.2.1 Implementasi program utama ekstraksi fitur (bagian pertama).....         | 46 |
| Kode Sumber 4.2.2 Implementasi program utama ekstraksi fitur (bagian kedua) .....          | 47 |
| Kode Sumber 4.2.3 Implementasi program utama ekstraksi fitur (bagian ketiga).....          | 48 |
| Kode Sumber 4.2.4 Implementasi fitur spectral centroid.....                                | 48 |
| Kode Sumber 4.2.5 Implementasi fitur spectral flux .....                                   | 49 |
| Kode Sumber 4.2.6 Implementasi fitur spectral rolloff .....                                | 49 |
| Kode Sumber 4.2.7 Implementasi fitur short time energy ....                                | 50 |
| Kode Sumber 4.2.8 Implementasi fitur zero crossing.....                                    | 50 |
| Kode Sumber 4.2.9 Implementasi ekstraksi fitur spectral flatness .....                     | 51 |
| Kode Sumber 4.2.10 Implementasi fungsi penghitungan nilai statistik (bagian pertama) ..... | 51 |
| Kode Sumber 4.2.11 Implementasi fungsi penghitungan nilai statistik (bagian kedua).....    | 52 |

*[Halaman ini sengaja dikosongkan]*



# **BAB I**

## **PENDAHULUAN**

Pada bab ini dibahas hal-hal yang mendasari Tugas Akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan Tugas Akhir.

### **1.1 Latar Belakang**

Musik memiliki karakteristik yang berbeda antara satu dengan lainnya. Penggunaan music berdasarkan genre telah digunakan pada kehidupan sehari-hari. Ketika terdapat sekumpulan berkas musik dalam jumlah yang besar, akan merepotkan untuk secara manual menggolongkannya kedalam kelompok-kelompok berdasarkan kesamaan karakteristik yang dimiliki tiap berkas. Melihat pesatnya industri musik sekarang dan jumlah berkas musik yang terus bertambah tiap waktu, maka pengelompokan berkas musik berdasarkan genre diperlukan.

Sebelum proses pengelompokan dilakukan, perlu terlebih dahulu melakukan proses ekstraksi fitur guna mendapatkan nilai-nilai atribut tiap berkas musik yang akan menjadi nilai pembeda satu berkas dengan berkas lainnya. Fitur yang diekstraksi haruslah komprehensif, padat, dan efektif. Berkas musik yang terkelompokkan dalam genre yang sama berarti mereka memiliki karakteristik yang mirip, karena musik-musik tersebut dimainkan oleh alat musik yang mirip, mempunyai pola irama, dan distribusi nada yang mirip. Fitur yang diekstraksi dalam pengerjaan tugas akhir ini adalah fitur *spectral centroid*, *spectral flux*, *spectral rolloff*, *spectral flatness*, dan *short time energy*. Penulis menggunakan referensi metode ekstraksi fitur dari Babu et al [1] sebagai referensi utama. Keseluruhan fitur tersebut akan diambil nilai rata-rata, deviasi standar, *kurtosis*, dan *skewness*nya sebagai atribut.

Adapun hasil yang diharapkan adalah tingginya tingkat akurasi hasil pengelompokan. Sehingga pendekatan metode

ekstraksi fitur yang digunakan pada Tugas Akhir ini dapat digunakan sebagai metode alternatif untuk ekstraksi fitur dan pengelompokkan musik berdasarkan genre.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Memahami konsep *Spectral centroid*, *spectral flux*, *spectral rolloff*, *spectral flatness*, dan *short time energy* sebagai fitur yang menjadi acuan proses klasifikasi.
2. Memahami konsep *Fourier transform* dan *Hamming window* sebagai metode untuk mengolah data sinyal pada musik.
3. Mengimplementasikan sistem yang dirancang untuk dapat melakukan ekstraksi fitur pada berkas musik.
4. Menyusun uji coba serta melakukan uji coba terhadap kumpulan berkas musik dengan menggunakan metode klasifikasi *Random Forest*.

## 1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Implementasi dilakukan dengan menggunakan perangkat lunak MATLAB 2015b.
2. Data yang digunakan untuk data latih dan data uji berupa berkas musik berformat .MP3 yang diambil dari website <http://freemusicarchive.org>.

## 1.4 Tujuan

Adapun beberapa tujuan dari pembuatan Tugas Akhir ini, yakni sebagai berikut:

1. Mengetahui peran *spectral centroid*, *spectral flux*, *spectral rolloff*, *spectral flatness*, dan *short time energy* sebagai fitur yang merepresentasikan musik berdasarkan genre.

2. Mengimplementasikan rancangan sistem yang dapat melakukan proses ekstraksi fitur pada musik.

### **1.5 Manfaat**

Dengan dibuatnya Tugas Akhir ini, hasil ekstraksi fitur musik kedepannya akan dapat digunakan untuk mengidentifikasi jenis genre pada musik.

### **1.6 Metodologi**

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.

Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Proposal Tugas Akhir yang diajukan memiliki gagasan yang sama dengan Tugas Akhir ini, yaitu mengelompokkan musik berdasarkan genre.

2. Studi literatur

Pada tahap ini dilakukan pencarian, pengumpulan, pembelajaran dan pemahaman informasi dan literatur yang diperlukan untuk mengolah berkas musik menjadi fitur yang dapat merepresentasikan musik. Informasi dan literatur didapatkan dari literatur buku dan sumber-sumber informasi lain yang berhubungan.

3. Perancangan perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari

sistem yang akan dibuat. Kemudian dilakukan desain suatu sistem dan desain proses-proses yang ada.

4. Implementasi perangkat lunak

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan dari tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan yang telah direncanakan. Implementasi proses ekstraksi fitur musik untuk klasifikasi berdasarkan genre menggunakan bantuan perangkat lunak MATLAB 2015b. Output yang diharapkan adalah fitur-fitur pada tiap berkas musik yang diolah, untuk kemudian digunakan sebagai data *training* dan data *testing* pada proses klasifikasi.

5. Pengujian dan evaluasi

Pada tahapan ini dilakukan uji coba pada data yang telah dikumpulkan. Pengujian dan evaluasi akan dilakukan dengan menggunakan MATLAB 2015b dan alat bantu Weka versi 3.8. Tahapan ini dimaksudkan untuk mengevaluasi kesesuaian data dan program serta mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

6. Penyusunan buku Tugas Akhir.

Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

## 1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:



**Bab I      Pendahuluan**

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

**Bab II     Dasar Teori**

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

**Bab III    Perancangan Perangkat Lunak**

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *pseudocode*.

**Bab IV    Implementasi**

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

**Bab V     Uji Coba Dan Evaluasi**

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

**Bab VI    Kesimpulan Dan Saran**

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini berisi penjelasan dasar teori yang berkaitan dengan algoritma yang diajukan pada pengimplementasian program. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

#### **2.1 Berkas MP3**

MPEG (Moving Picture Expert Group)-1 atau lebih dikenal dengan MP3 merupakan salah satu format berkas pengodean suara. Teknologi ini dikembangkan oleh Karlheinz Brandenburg, insinyur Institut Fraunhofer di Jerman. MP3 terdiri dari banyak sekali frame, dimana setiap frame mengandung sebagian detik dari data audio yang berguna, yang siap dikonstruksi ulang oleh *decoder* [2].

Kompresi yang dilakukan oleh model pengkodean mp3 tidak mempertahankan bentuk asli dari sinyal input. Melainkan dengan cara menghilangkan suara-suara yang keberadaannya tidak memberi pengaruh banyak terhadap system pendengaran manusia. Proses penghilangan ini dilakukan dengan menggunakan model dari sistem pendengaran manusia dan menentukan bagian yang terdengar bagi sistem pendengaran manusia. Selanjutnya sinyal input yang memiliki domain waktu dibagi-bagi menjadi blok-blok dan ditransformasi menjadi domain frekuensi. Kemudian model dari sistem pendengaran manusia dibandingkan dengan sinyal input dan dilakukan proses pemfilteran yang menghasilkan sinyal dengan frekuensi yang secara efisien dapat terdengar oleh sistem pendengaran manusia. MP3 menghilangkan komponen-komponen suara yang tidak terdengar oleh manusia dengan mengurangi jumlah bit yang diperlukan. Rentang bit rate pada format MP3 adalah dari 32 kbits/detik sampai 320 kbits/detik [2].

Adapun data yang tersimpan pada setiap berkas MP3 adalah nilai amplitudo pada tiap bit dan nilai bit *rate* yang menunjukkan banyak bit dalam satu detik pada berkas. Data tersebut yang nantinya diambil untuk diolah dan diambil fiturnya.

Berkas MP3 yang digunakan pada Tugas Akhir ini diambil dari website <http://freemusicarchive.org>. Website ini menyediakan data berkas musik yang dapat diunduh secara gratis dan legal dengan dilengkapi fitur penyaringan pencarian berkas musik berdasarkan genre [3].

## 2.2 Ekstraksi fitur

Ekstraksi fitur merupakan suatu proses pengambilan ciri atau karakteristik dari suatu data yang dapat merepresentasikan informasi penting untuk selanjutnya dimanfaatkan dalam kebutuhan proses analisa data maupun klasifikasi. Dalam Tugas Akhir ini, data yang digunakan pada saat proses ekstraksi fitur berupa data file *audio MP3*. Dan fitur yang diekstraksi adalah *spectral centroid*, *spectral flux*, *spectral rolloff*, *spectral flatness*, *short time energy*, dan *zero crossing* yang akan dijelaskan pada subbab selanjutnya [1].

Untuk dapat melakukan rangkaian tahapan ekstraksi fitur, diperlukan pengambilan data nilai amplitudo tiap bit dan nilai banyak bit tiap detik pada berkas MP3 yang diolah. Dalam dunia digital, file *audio MP3* memiliki informasi yang dinamakan *sample rate*. *Sample rate* merupakan banyaknya jumlah data yang dibawa tiap detik dari sinyal yang diterima dan biasa dinyatakan dalam satuan Hz. Pada umumnya *sample rate* pada file audio musik yang banyak dikonsumsi masyarakat bernilai 44100 Hz, yang berarti setiap detiknya file audio tersebut menyimpan 44100 data. Alasan kenapa banyak file *audio* menggunakan nilai 44100 Hz sebagai *sample rate*, karena kemampuan manusia untuk menangkap frekuensi suara adalah dari 20 Hz hingga 20 kHz, sehingga nilai *sample rate* yang paling efisien untuk digunakan adalah 44100 Hz [3].

Dari informasi nilai *sample rate* dan data sampel yang ada pada file *audio* yang diolah, proses perhitungan nilai fitur pada Tugas Akhir ini pertama kali dilakukan dengan membagi data sampel menjadi beberapa bagian yang sama besar atau disebut dengan *frame*. Besar nilai panjang *frame* disesuaikan dengan informasi nilai *sample rate* pada file *audio* yang diolah. Kemudian pada tiap *frame* diambil nilai fiturnya, dan selanjutnya dihitung nilai rata-rata, standard deviasi, *skewness*, dan *kurtosis* dari keseluruhan nilai masing-masing fitur yang didapat pada perhitungan tiap *frame* [1].

Perhitungan nilai fitur *spectral centroid*, *spectral flux*, *spectral rolloff*, *spectral flatness*, *short time energy*, dan *zero crossing* dilakukan di tiap *frame* yang telah terbentuk. Sebagai contoh, pada pengolahan satu berkas mp3 yang berjudul *Songbird.mp3*, dengan data amplitudo sebanyak 10362412 buah data dan panjang *frame* yang ditetapkan sebesar  $0.01 * \text{sample rate}$ , dan nilai *sample rate* pada berkas mp3 tersebut adalah 44100. Sehingga total perhitungan nilai fitur untuk berkas mp3 tersebut adalah sebanyak 23497 data nilai fitur.

### 2.2.1 Spectral Centroid

*Spectral Centroid* merupakan fitur yang merepresentasikan titik pada spektrum, dimana sebagian besar energi terpusat di titik tersebut [1]. Selain itu, *spectral centroid* sendiri adalah sebuah konsep yang diambil dari pengertian musik dan psikoakustik, merupakan titik keseimbangan spektrum yang digunakan untuk mengukur ketajaman spektral dan sering dikaitkan dengan tingkat kejelasan spektral, dimana semakin tinggi nilai *spectral centroid* maka suara yang ada akan semakin tajam dan jelas. Kecerahan atau ketajaman suara dideskripsikan sebagai lawan dari tingkat kebisingan pada suara. Secara umum *spectral centroid* memiliki persamaan yang didefinisikan pada Persamaan 2.1.

$$C_t = \frac{\sum_{n=1}^N M_t[n] * n}{\sum_{n=1}^N M_t[n]} \quad (2.1)$$

dimana  $n$  adalah nilai indeks bit,  $C_t$  adalah nilai *spectral centroid* pada frame ke- $t$ ,  $N$  adalah jumlah total bit yang ada,  $M_t[n]$  adalah nilai dari transformasi Fourier pada frame ke  $t$  dan indeks bit ke  $n$  [1].

### 2.2.2 Spectral Flux

*Spectral Flux* merupakan fitur yang mengukur seberapa cepat energi dari sinyal berubah. Perhitungan *spectral flux* dilakukan dengan membandingkan energi sinyal pada sebuah frame dengan frame sebelumnya sehingga dapat mengukur perbedaan spektral dari frame ke frame dan dapat mengkarakteristikan perubahan bentuk spektrum [1]. Secara umum *spectral flux* memiliki persamaan yang didefinisikan pada Persamaan 2.2.

$$F_t = \sum_{n=1}^N (N_t[n] - N_{t-1}[n])^2 \quad (2.2)$$

dimana  $F_t$  adalah nilai spectral flux pada frame ke- $t$ ,  $N$  adalah banyak bit data pada frame ke- $t$ , dan  $N_t[n]$  adalah hasil normalisasi besaran nilai transformasi fourier pada frame ke- $t$  dan indeks bit ke- $n$ . Hasil normalisasi didapatkan dari data besaran nilai transformasi fourier pada frame ke- $t$  ( $M_t$ ) dibagi dengan nilai maksimal pada data tersebut.

### 2.2.3 Spectral Rolloff

*Spectral rolloff* merupakan fitur yang menunjukkan nilai indeks bit ketika jumlah komulatif besaran energi mencapai 85% dari total energi sinyal pada frame. Hasil perhitungan *spectral rolloff* dapat digunakan untuk mengukur apakah suara termasuk dalam *voiced speech* ataupun *unvoiced speech*. *Voiced speech* pada umumnya memiliki nilai spectral rolloff yang lebih kecil apabila

dibandingkan dengan *unvoiced speech* [1]. Sama seperti halnya dengan *spectral centroid*, *spectral rolloff* digunakan untuk mengukur ketajaman spektral. Nilai *spectral rolloff* juga untuk mengukur keasimetrisan bentuk spektral. Nilai *spectral rolloff* akan semakin tinggi untuk distribusi yang lebih asimetris. Secara umum *spectral rolloff* memiliki persamaan yang didefinisikan pada Persamaan 2.3

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 \sum_{n=1}^N M_t[n] \quad (2.3)$$

dimana  $R_t$  adalah nilai indeks bit data sinyal ketika jumlah komulatif energi sinyal mencapai 85% dari jumlah total energi pada *frame*,  $M_t[n]$  merupakan besar energi sinyal hasil transformasi *fourier* pada *frame* ke- $t$  indeks ke- $n$  dan  $N$  merupakan jumlah total bit yang ada pada *frame*  $t$ . Nilai fitur *spectral rolloff* didapatkan dengan mencari nilai  $R_t$  pada Persamaan 2.3 pada *frame*  $t$  yang diolah.

#### 2.2.4 Short Time Energy

*Short time energy* merupakan fitur yang menunjukkan nilai untuk mengukur kecenderungan apakah suara termasuk kedalam kategori *voiced speech* atau *unvoiced speech*. Nilai *short time energy* dari suara *voiced speech* akan lebih besar apabila dibandingkan dengan suara *unvoiced speech* [1]. Secara umum, persamaan dari *short time energy* dapat dilihat pada Persamaan 2.4.

$$E_t = \sum_{m=1}^N [x(m)w(m)]^2 \quad (2.4)$$

Dimana  $E_t$  adalah nilai fitur *short time energy*,  $x(m)$  merupakan besaran hasil transformasi *fourier* dari data sinyal indeks ke- $m$ ,  $w(m)$  adalah data sinyal pada *window* dengan panjang  $N$ , dan  $N$  merupakan jumlah total bit yang ada pada *frame*  $t$ .

### 2.2.5 Zero Crossing

*Zero crossing* merupakan fitur yang nilainya menunjukkan jumlah berapa kali nilai amplitudo pada bit data sinyal melewati nilai 0. Suara yang tergolong *unvoiced speech* apabila dibandingkan dengan suara *voiced speech* seringkali memiliki nilai *zero crossing* yang lebih besar [1]. Sebuah *zero crossing* terjadi saat sampel data dalam sebuah sinyal digital mempunyai perbedaan tanda. Secara umum, persamaan dari *zero crossing* dapat dilihat pada Persamaan 2.5.

$$Z_t = \frac{1}{2} \sum_{n=1}^N |\text{sign}(x[n] - \text{sign}(x[n-1]))| \quad (2.5)$$

Dimana  $Z_t$  merupakan nilai fitur *zero crossing*,  $x[n]$  adalah nilai amplitudo pada data indeks ke- $n$ ,  $N$  merupakan jumlah total bit yang ada pada *frame*  $t$ , dan *sign* adalah fungsi, yang akan bernilai 1 apabila positif dan bernilai 0 apabila negatif.

### 2.2.6 Spectral Flatness

*Spectral flatness* merupakan fitur yang menunjukkan karakteristik dari sebuah spektrum audio yang mana sering direpresentasikan dalam satuan desibel [1]. Secara umum *spectral flatness* memiliki persamaan yang didefinisikan pada Persamaan 2.6

$$F_t = \frac{\exp\left(\frac{1}{N} \sum_{m=0}^{N-1} \ln x(m)\right)}{\frac{1}{N} \sum_{m=0}^{N-1} x(m)} \quad (2.6)$$

Dimana  $F_t$  merupakan nilai fitur *spectral flatness*,  $x[m]$  adalah nilai amplitudo pada data indeks ke- $m$ ,  $N$  merupakan jumlah total bit yang ada pada *frame*  $t$ , dan  $m$  adalah indeks data sinyal [1].



### 2.2.7 Discrete Fourier Transformation (DFT)

Discrete Fourier Transformation (DFT) memegang peranan penting dalam pemrosesan sinyal, karena tujuan utama dari DFT adalah mengubah sinyal dengan domain waktu ke dalam domain frekuensi.

Secara umum, persamaan dari *DFT* dapat dilihat pada Persamaan 2.7.

$$Y[k] = \sum_{j=0}^{n-1} X[j] \omega_n^{j \cdot k} \quad (2.7)$$

Dimana  $Y[k]$  merupakan array data sinyal dengan domain frekuensi, dan  $X[j]$  merupakan array data sinyal dengan domain waktu. Dengan  $0 \leq k \leq n$  dan  $\omega_n = e^{-j2\pi/N}$  [5]. Konsep utama dari DFT adalah, mengubah data sinyal dari domain waktu menjadi ke dalam domain frekuensi.

### 2.2.8 Fast Fourier Transform (FFT)

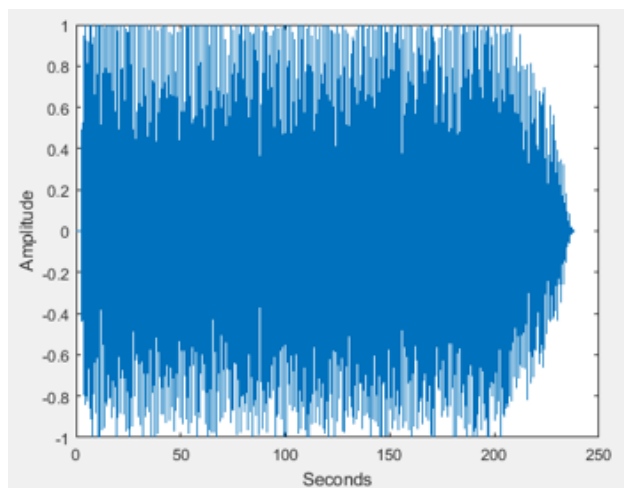
Fast Fourier Transform atau yang biasa disingkat FFT adalah metode pengerjaan cepat dari Discrete Fourier Transform (DFT). FFT merupakan teknik perhitungan transformasi fourier yang dirumuskan oleh J.W. Cooley dan J.W. Turkey pada tahun 1960. Istilah *fast* digunakan oleh karena formulasi FFT ini memiliki komputasi jauh lebih cepat dibandingkan dengan transformasi fourier. FFT menerapkan metode perhitungan yang lebih cepat daripada DFT. Apabila perhitungan formulasi DFT secara langsung akan membutuhkan operasi aritmatika sebanyak  $(N^2)$ , sedangkan perhitungan dengan formulasi FFT akan membutuhkan operasi aritmatika sebanyak  $O(N \log N)$  [5].

Untuk mengerjakan perhitungan DFT pada alat bantu Matlab, digunakan fungsi *fft(y,N)* dimana  $y$  adalah sinyal yang akan ditransformasikan, dan  $N$  adalah nilai ukuran matriks keluaran fungsi. Apabila  $N$  tidak dideklarasikan maka nilainya akan sama dengan ukuran  $x$ . Hasil keluaran fungsi *fft()* memiliki

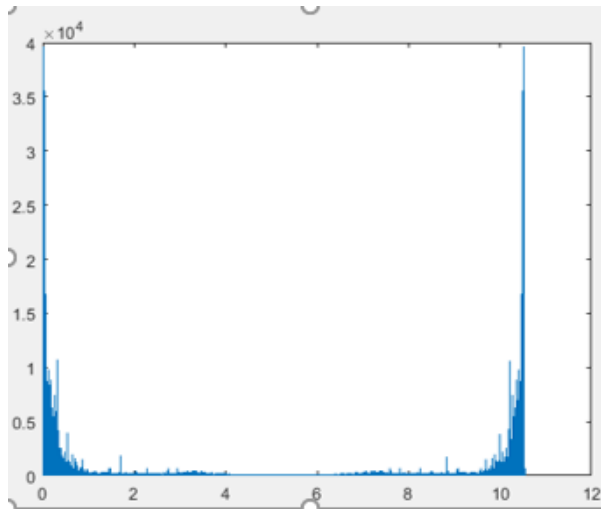
panjang  $N$ , yang mana indeks 1 sampai dengan  $N/2$  simetri dengan  $N/2$  sampai dengan  $N$  [5].

Gambar 2.2.1 menunjukkan grafik sinyal suara digital pada berkas mp3 lagu berjudul “*Stormy Blues*” oleh Arne Bang Huseby yang ditampilkan dengan menggunakan alat bantu Matlab, sumbu x menunjukkan waktu (detik) dan sumbu y menunjukkan amplitudo. Agar didapatkan hasil transformasi fourier pada data sinyal yang digambarkan pada Gambar 2.2.1, maka dilakukan fungsi  $fft(y)$  pada matlab dan menghasilkan grafik seperti pada Gambar 2.2.2, dimana sumbu x menunjukkan indeks frekuensi dan sumbu y menunjukkan besaran nilai transformasi fourier.

Prinsip dasar dari formulasi Fast Fourier Transform (FFT) adalah menguraikan perhitungan  $N$ -titik DFT menjadi perhitungan DFT dengan ukuran yang lebih kecil dan memanfaatkan periodisitas dan simetri dari bilangan kompleks  $\omega_n^{j.k}$  pada formulasi DFT yang ditunjukkan pada Persamaan 2.7.



**Gambar 2.2.1** Sinyal suara digital pada berkas mp3 lagu berjudul “*Stormy Blues*” oleh Arne Bang Huseby



**Gambar 2.2.2** Hasil transformasi fourier sinyal suara digital pada berkas mp3 lagu berjudul “*Stormy Blues*” oleh Arne Bang Huseby

### 2.2.9 *Hamming Window*

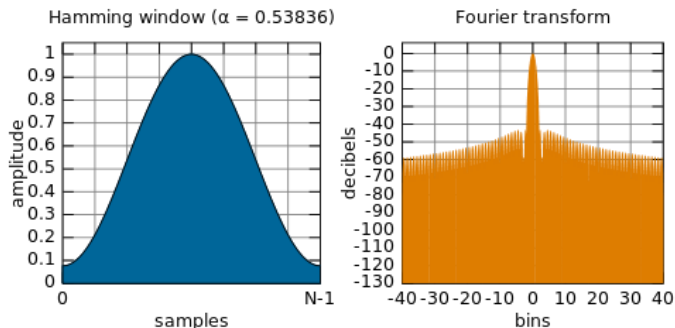
*Hamming window* atau jendela *hamming* merupakan salah satu metode *windowing* yang sering digunakan untuk memfilter nilai pada data sinyal. Secara matematis *hamming window* terdiri dari satu siklus kosinus, yang menaikkan dan menghimpitkan sehingga bentuk gelombang turun dan mempunyai nilai puncak. Metode ini pertama kali diusulkan oleh ilmuwan matematika berkebangsaan Amerika bernama Richard W. Hamming untuk menyederhanakan perhitungan.

$$w(n) = 0,54 - 0,46 * \left(1 - \cos\left(2\pi \frac{n}{N}\right)\right) \quad (2.8)$$

Rumus *hamming window* dapat dilihat pada Persamaan 2.7. Penggunaan *window* dalam pemrosesan data sinyal,

diperlukan karena dapat mengurangi dampak dari batas diskontinuitas (*boundary of discontinues*). Diskontinuitas berada pada awal dan akhir tiap frame. *Window* berupaya untuk mengurangi kerusakan spectrum dengan cara meruncingkan sinyal dengan menset nol pada awal dan akhir tiap *frame*. *Window* diaplikasikan dalam sinyal domain waktu sebelum digunakan pada proses FFT [6]. Grafik spectrum hasil *hamming window* dapat diperlihatkan pada Gambar 2.2.1.

Pada Tugas Akhir ini, *Hamming Window* digunakan ketika tahap pra proses, yaitu sebelum dilakukan perhitungan nilai fitur pada *frame* yang dibentuk.



**Gambar 2.2.3** Spektrum dengan *hamming window*

### 2.2.10 Mean

*Mean* atau biasa disebut rata-rata merupakan suatu nilai bilangan atau ukuran statistik yang menunjukkan dimana sekumpulan data terpusat. Perhitungan nilai *mean* dilakukan dengan menjumlahkan seluruh nilai data suatu kelompok sampel, kemudian dibagi dengan jumlah sampel tersebut.

Secara umum, persamaan dari *mean* dapat dilihat pada Persamaan 2.8.

$$\text{Mean}(\mu) = \frac{1}{N} \sum_{n=1}^N X_n \quad (2.8)$$

Dimana N adalah jumlah sampel dan  $X_n$  adalah nilai sampel ke-n pada kelompok sampel X [1].

### 2.2.11 Deviasi Standar

Istilah deviasi standar pertama kali diperkenalkan oleh Karl Pearson pada tahun 1894, dalam bukunya yang berjudul “*On the dissection of asymmetrical frequency curves*”. Deviasi standar atau juga biasa disebut dengan simpangan baku merupakan nilai statistik yang digunakan untuk menentukan bagaimana sebaran data dalam sampel, dan seberapa dekat titik data individu ke nilai *mean* pada sampel.

Apabila nilai sebuah perhitungan simpangan baku dari kumpulan data sama dengan nol, maka ini menunjukkan bahwa nilai-nilai dalam kumpulan data tersebut adalah sama besar. Sebuah nilai simpangan baku yang lebih besar, akan menunjukkan keseragaman data sampel yang lebih besar. Simpangan baku merupakan bilangan tak negatif, dan memiliki satuan yang sama dengan data.

Secara umum, persamaan dari simpangan baku dapat dilihat pada Persamaan 2.9.

$$\text{Std}(\sigma) = \frac{1}{N} \sum_{n=1}^N (X_n - \mu)^2 \quad (2.9)$$

Dimana N adalah jumlah sampel dan  $X_n$  adalah nilai sampel ke-n pada kelompok sampel X [1].

### 2.2.12 Skewness

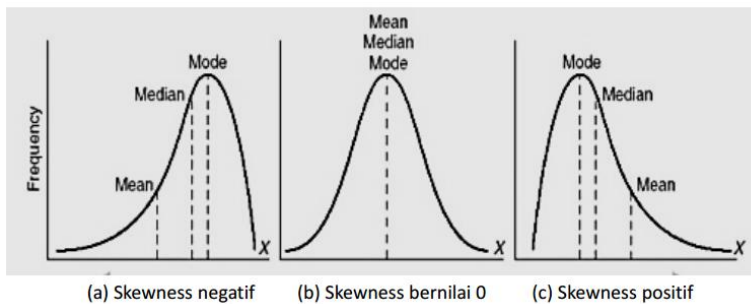
*Skewness* atau disebut juga dengan ukuran kemiringan merupakan suatu bilangan yang dapat menunjukkan miring atau tidaknya bentuk kurva suatu distribusi frekuensi [1]. *Skewness* adalah derajat ketidaksamaan suatu distribusi. Secara perhitungan,

*skewness* adalah momen ketiga terhadap *mean*. Kurva dengan distribusi normal memiliki nilai *skewness* 0 yang berarti data terdistribusi secara simetrik, dan grafik kurva yang dihasilkan tidak condong ke kiri maupun ke kanan. Nilai *skewness* akan lebih dari 0 pada data yang terdistribusi positif, grafik kurva yang dihasilkan akan miring ke kiri. Dan sebaliknya, nilai *skewness* bernilai kurang dari 0 apabila data yang ada terdistribusi negative, grafik kurva yang dihasilkan akan miring ke kanan.

Secara umum, persamaan yang digunakan untuk mencari nilai *skewness* dapat dilihat pada Persamaan 2.10.

$$Skewness = \frac{\sum_{n=1}^N (X_n - \mu)^3}{(N-1)\sigma^3} \quad (2.10)$$

Dimana N adalah jumlah sampel dan  $X_n$  adalah nilai sampel ke-n pada kelompok sampel X [1]. Perbedaan grafik kurva untuk *skewness* positif, normal, dan *skewness* negative ditunjukkan pada Gambar 2.2.4.



**Gambar 2.2.4** Perbedaan grafik kurva berdasarkan nilai *skewness*.

### 2.2.13 Kurtosis

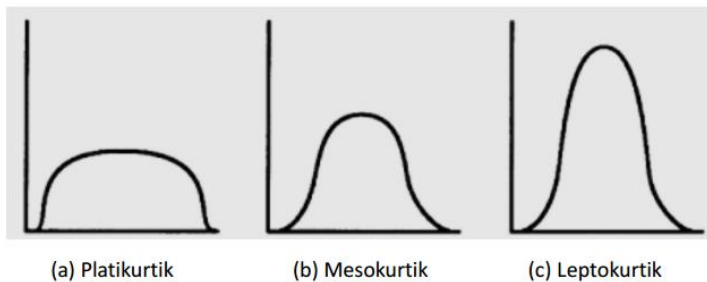
Kurtosis merupakan pengukuran tinggi rendahnya grafik distribusi frekuensi, terutama berkenaan dengan konsentrasi nilai

yang mendekati rata-rata jika dibandingkan dengan distribusi normal [1]. Kurtosis atau juga disebut dengan ukuran keruncingan adalah suatu bilangan yang dapat menunjukkan runcing tidaknya bentuk kurva distribusi frekuensi. Kurtosis adalah derajat keruncingan suatu distribusi yang biasanya diukur relative terhadap distribusi normal. Kurva yang lebih runcing dari distribusi normal dinamakan leptokurtik, yang lebih datar dinamakan dengan platikurtik, dan pada distribusi normal disebut dengan mesokurtik. Kurtosis dihitung dari momen keempat terhadap *mean*.

Secara umum, persamaan yang digunakan untuk mencari nilai kurtosis dapat dilihat pada Persamaan 2.11.

$$Kurtosis = \frac{\sum_{n=1}^N (X_n - \mu)^4}{(N-1)\sigma^4} - 3 \quad (2.11)$$

Dimana N adalah jumlah sampel dan  $X_n$  adalah nilai sampel ke-n pada kelompok sampel X [1]. Perbedaan bentuk grafik kurva leptokurtic, platikurtik, dan mesokurtik ditunjukkan pada Gambar 2.2.5.



**Gambar 2.2.5** Perbedaan kategori gravik kurva berdasarkan nilai kurtosis

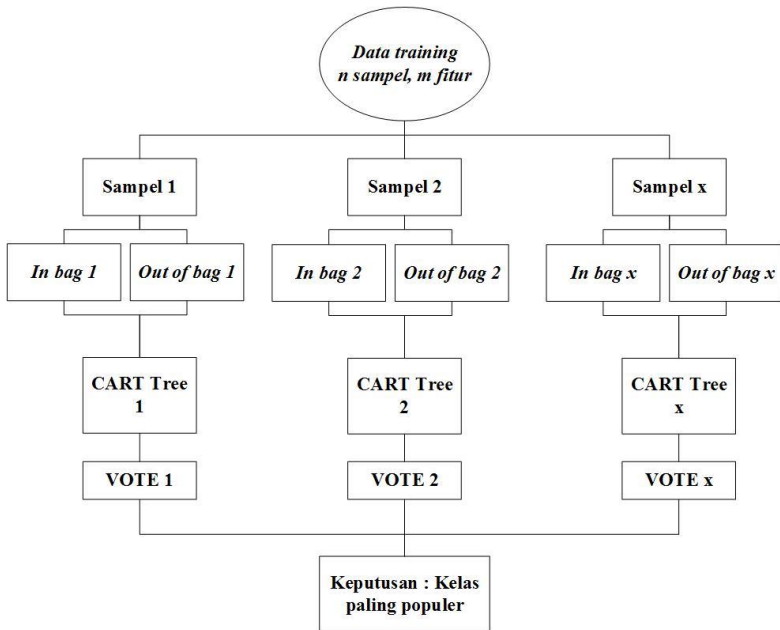
## 2.3 Klasifikasi Random Forest

*Random Forest* merupakan metode klasifikasi yang pertama kali dikembangkan oleh Leo Breiman ilmuwan asal Amerika Serikat. Metode klasifikasi *Random Forest* terdiri dari beberapa pohon keputusan sebagai *classifier*. Kelas yang dihasilkan dari proses klasifikasi ini diambil dari kelas terbanyak yang dihasilkan oleh pohon-pohon keputusan yang ada pada *Random Forest*. *Random Forest* tidak berkecenderungan untuk *overfit* dan dapat melakukan proses dengan cepat, sehingga memungkinkan untuk memproses *tree* sebanyak yang diinginkan oleh pengguna [7].

Dalam pembentukan *tree*, algoritma *Random Forest* melakukan training terhadap sampel data. Pengambilan sampel dilakukan dengan cara *sampling with replacement*. Sebanyak sepertiga dari sampel akan digunakan sebagai data *out of bag* dan sisanya sebagai data *in bag*. Data *out of bag* digunakan untuk mengestimasi error dan menentukan *variable importance*. Variabel yang akan digunakan untuk menentukan pemisahan, ditentukan secara acak. Dana data *in bag* digunakan untuk diolah kedalam pohon keputusan. Setelah seluruh *tree* terbentuk, maka proses klasifikasi akan berjalan. Penentuan kelas dilakukan dengan cara *voting* dari masing-masing *tree*, kelas dengan jumlah *vote* terbanyak akan menjadi kelas terpilih. Proses klasifikasi dengan metode *Random Forest* ini ditunjukkan pada Gambar 2.3.1.

Pada metode klasifikasi *Random Forest*, terdapat beberapa parameter yang memberikan pengaruh terhadap nilai akurasi hasil klasifikasinya. Parameter-parameter tersebut adalah ukuran tiap *bag*, banyak fitur yang digunakan, banyak iterasi yang dilakukan. Yang dimaksud dengan ukuran tiap *bag* adalah ukuran banyak data berupa persentase dari total keseluruhan jumlah data *training* yang dilakukan oleh tiap pohon keputusan yang dibentuk [8].





**Gambar 2.3.1** Proses klasifikasi *Random Forest*.

*[Halaman ini sengaja dikosongkan]*

### BAB III PERANCANGAN

Pada bab ini akan dijelaskan perancangan perangkat lunak pengelompokan berkas musik berdasarkan genre. Perancangan akan dibagi menjadi dua proses utama, yaitu:

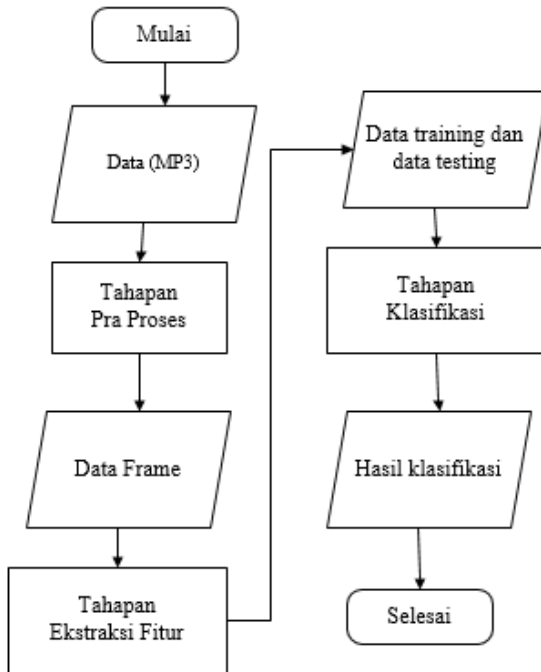
1. Pra Proses
2. Ekstraksi fitur.
3. Klasifikasi.

Di dalam masing-masing proses utama akan dibagi menjadi proses-proses kecil yang terlibat di dalamnya. Pada proses ekstraksi fitur, dilakukan proses pembacaan berkas MP3 yang menjadi data input untuk diambil nilai amplitudo tiap bit nya dan nilai frekuensi *rate*. Kemudian dilakukan perhitungan fitur pada tiap *frame* pada data input. Gambar 2.3.1 dijelaskan mengenai gambaran umum proses dari bab ini.

Sebelum dilakukan ekstraksi fitur, terdapat pra proses yang dilakukan untuk membagi data-data sinyal ke dalam frame-frame waktu dengan panjang durasi tiap *frame* dalam satuan waktu adalah 10 mikro detik. Banyak data yang terdapat pada rentang waktu 10 mikro detik tergantung pada *sample rate* pada berkas musiknya. Definisi dari *sample rate* adalah banyaknya data yang ada pada berkas musik tiap detiknya.

Ekstraksi fitur pada Tugas Akhir ini, dilakukan pada tiap *frame-frame* yang dihasilkan sebelumnya saat pra proses. Kemudian diambil nilai rata-rata, deviasi standar, *skewness*, dan kurtosis pada masing-masing vektor nilai hasil perhitungan nilai fitur yang dihasilkan. Jenis fitur yang diekstraksi adalah *spectral centroid*, *spectral flux*, *spectral rolloff*, *spectral flatness*, *short time energy*, dan *zero crossing*.

Adapun pada bab ini akan dijelaskan gambaran umum mengenai metode dari setiap proses utama dalam bentuk bagan yang kemudian dilanjutkan dengan penjelasan lebih detail dalam bentuk *pseudocode*.



**Gambar 2.3.1 Diagram alir keseluruhan proses**

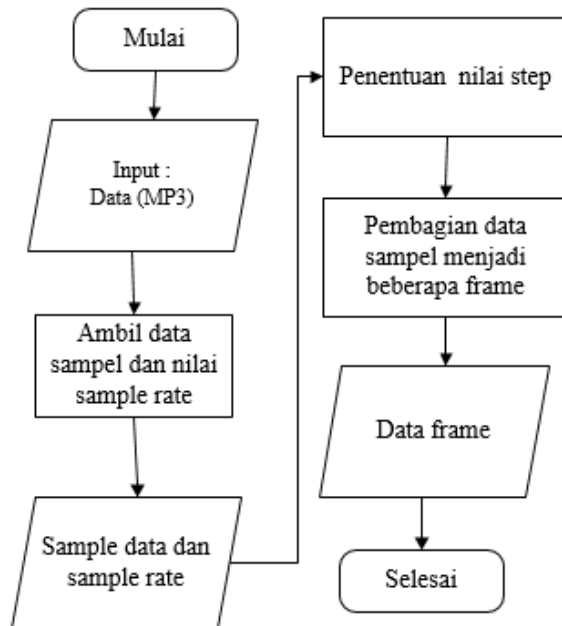
### 3.1 Pra Proses

Pra proses merupakan proses yang dilakukan di awal pada aplikasi ini. Masukkan dari tahapan ini berupa berkas mp3 yang telah diketahui jenis genrenya, dan keluarannya berupa data sampel, sample rate, dan data sampel yang telah dibagi-bagi kedalam bagian lebih kecil yang disebut dengan *frame*.

Pada tahap pra proses dilakukan proses pengambilan data sampel dari berkas musik yang akan diolah, kemudian dilanjutkan

dengan pembagian data sampel kedalam bagian-bagian yang lebih kecil dengan ukuran sama yang biasa disebut dengan *frame*. Panjang *frame* ditentukan oleh nilai sampel rate yang didapat pada proses pembacaan berkas MP3.

Penentuan nilai step dilakukan pada tahapan ini guna menentukan tipe data fragmen yang dibentuk, apakah ingin dikehendaki data bersifat overlap dengan data fragmen tetangganya, ataukah tidak. Hal ini akan digunakan sebagai skenario pada saat tahapan uji coba. Diagram alir proses pengerjaan pada tahapan Pra Proses ditunjukkan pada Gambar 3.1.1.



**Gambar 3.1.1 Diagram alir tahapan pra proses**

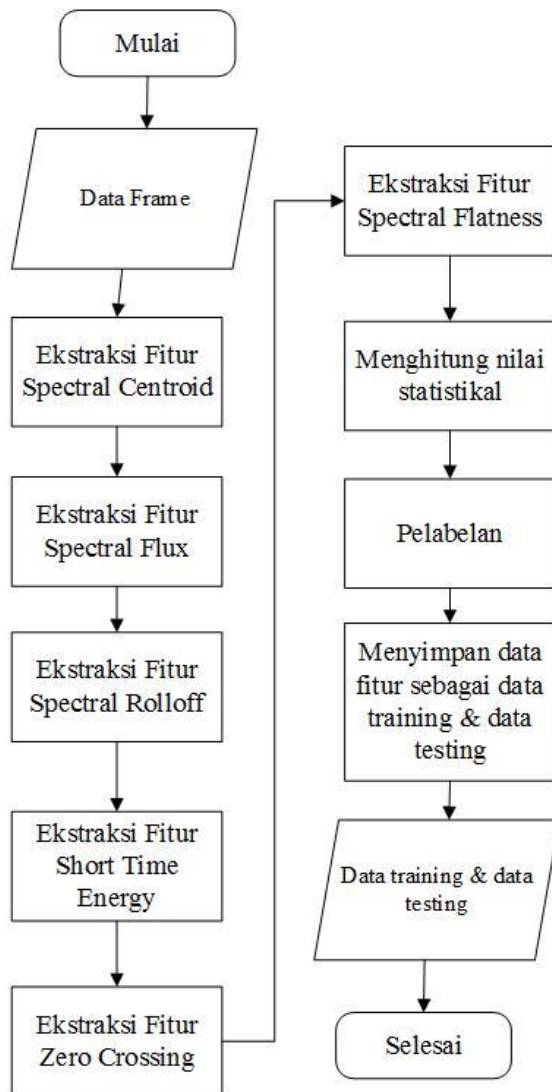
### 3.2 Ekstraksi Fitur

Ekstraksi fitur merupakan proses awal pada aplikasi ini. Terdapat tujuh subproses di dalam tahapan ekstraksi fitur, yang masing-masing merupakan proses perhitungan fitur. Sebelum dilakukan perhitungan untuk mengekstraksi fitur, program mengambil *sample data* dan *sample rate* terlebih dahulu pada berkas musik yang akan diolah.

Dari *sample data* dan *sample rate* yang didapat, proses perhitungan nilai fitur pada Tugas Akhir ini dilakukan pada tiap *frame* dengan panjang *frame* ditentukan sebanyak  $0.01 \times \text{sample rate}$  data yang sama dengan banyak *sample data* pada berkas musik selama 10 mikro detik. Selisih panjang data yang dilewati antar posisi awal suatu *frame* dengan posisi awal *frame* selanjutnya dinamakan dengan *step*.

Setelah perhitungan nilai enam fitur pada tiap *frame* selesai dilakukan, maka dihitung empat nilai statistikalnya untuk masing-masing fitur. Empat nilai statistikal yang dihitung pada tiap fitur adalah *mean*, deviasi standar, *skewness*, dan kurtosis. Sehingga output dari tahapan ini menghasilkan 24 atribut yang digunakan sebagai data *training* dan data *testing* pada tahapan klasifikasi. Penjelasan mengenai masing-masing nilai statistikal yang digunakan ini telah dijabarkan pada subbab 2.2.9 hingga 2.2.12.

Untuk penjelasan lebih detil mengenai tahapan ekstraksi fitur, akan dijelaskan dalam bentuk *pseudocode*. *Pseudocode* proses utama ini terdiri dari suatu program utama yang memiliki tujuan untuk memanggil beberapa program kecil lainnya yang berisi program yang lebih kecil untuk dijalankan. Pada Tabel 3.2.1 hingga Tabel 3.2.3 akan dijelaskan variabel-variabel yang akan digunakan pada *pseudocode*. Sedangkan pada Tabel 3.2.4 akan dijelaskan fungsi-fungsi yang akan dipakai di dalam *pseudocode*. Diagram alir proses pengerjaan ekstraksi fitur ditunjukkan pada Gambar 3.2.1.



**Gambar 3.2.1 Diagram alir tahap ekstraksi fitur**

**Tabel 3.2.1** Daftar Variabel yang Digunakan Pada Pseudocode Ekstraksi Fitur (Bagian Pertama)

| No | Nama Variabel | Tipe    | Penjelasan   |
|----|---------------|---------|--|
| 1  | curPos        | integer | Variabel untuk menyimpan indeks pada data <i>y</i> yang menjadi indeks pertama pada <i>frame</i> . |
| 2  | features      | double  | Keluaran pada program utama yang menyimpan nilai fitur-fitur pada berkas musik yang diolah.        |
| 3  | features      | double  | Keluaran pada program utama yang menyimpan nilai fitur-fitur pada berkas musik yang diolah.        |
| 4  | frame         | double  | Hasil kali antara hamming window dengan data amplitudo pada <i>frame</i> yang sedang diolah        |
| 5  | frameLength   | integer | Ukuran panjang <i>frame</i> .  |
| 6  | Fs            | integer | Nilai <i>sample rate</i> pada berkas musik yang akan diolah.                                       |
| 7  | H             | double  | Hamming window sebagai filter yang digunakan saat mengolah data musik                              |
| 8  | i             | integer | Indeks bit data pada <i>frame</i> .  |
| 9  | iRolloff      | double  | Nilai indeks pada data dalam <i>frame</i> yang diolah menggunakan fungsi <i>FRolloff</i>           |
| 10 | Mn            | double  | Nilai hasil transformasi <i>fourier</i> pada data <i>frame</i> .                                   |
| 11 | Na            | double  | Normalisasi hasil transformasi <i>fourier</i> pada <i>frame</i> yang sedang diolah.                |
| 12 | Nb            | double  | Normalisasi hasil transformasi <i>fourier</i> pada <i>frame</i> sebelumnya.                        |



**Tabel 3.2.2** Daftar Variabel yang Digunakan Pada Pseudocode Ekstraksi Fitur (Bagian Kedua)

| No | Nama Variabel    | Tipe    | Penjelasan  |
|----|------------------|---------|---|
| 13 | ShortTimeEnergy  | double  | Nilai fitur <i>short time energy</i> pada tiap data <i>frame</i>                  |
| 14 | SpectralCentroid | double  | Nilai <i>spectral centroid</i> pada tiap data <i>frame</i>                        |
| 15 | SpectralFlatness | double  | Nilai fitur <i>spectral flatness</i> pada tiap data <i>frame</i>                  |
| 16 | SpectralFlux     | double  | Nilai fitur spectral flux pada tiap data frame                                    |
| 17 | SpectralFlux     | double  | Nilai <i>spectral flux</i> pada tiap data   |
| 18 | SpectralRolloff  | double  | Nilai fitur <i>spectral rolloff</i> pada tiap data <i>frame</i>                   |
| 19 | step             | integer | Selisih antara data pertama dari frame dengan data pertama pada frame selanjutnya |
| 20 | x                | double  | Nilai amplitudo dalam frame pada data musik yang diolah.                          |
| 21 | y                | double  | Nilai amplitudo tiap bit pada berkas musik yang diolah.                           |
| 22 | ZeroCrossing     | double  | Nilai fitur <i>zero crossing</i> pada tiap data <i>frame</i>                      |

**Tabel 3.2.3** Daftar Fungsi yang Digunakan Pada Pseudocode program Ekstraksi Fitur (Bagian Pertama)

| No. | Nama Fungsi | Penjelasan   |
|-----|-------------|--|
| 1   | audioread   | Membaca nilai amplitudo tiap bit dan nilai sample rate pada berkas lagu terpilih                                 |
| 2   | fft         | Menghitung nilai <i>discrete fourier transform</i> pada data menggunakan algoritma <i>fast fourier transform</i> |

**Tabel 3.2.4** Daftar Fungsi yang Digunakan Pada Pseudocode program Ekstraksi Fitur (Bagian Kedua)

|    |                                |   |
|----|--------------------------------|---|
| 3  | <code>floor</code>             | Mengembalikan nilai pembulatan kebawah dari suatu nilai                                 |
| 4  | <code>fprintf</code>           | Menuliskan data pada file   |
| 5  | <code>FShortTimeEnergy</code>  | Menghitung nilai standard deviasi dari suatu vektor                                     |
| 6  | <code>FSpectralCentroid</code> | Menghitung nilai fitur <i>short time energy</i> pada tiap data <i>frame</i>             |
| 7  | <code>FSpectralFlatness</code> | Menghitung nilai <i>spectral centroid</i> pada tiap data <i>frame</i>                   |
| 8  | <code>FSpectralFlux</code>     | Menghitung nilai fitur <i>spectral flatness</i> pada tiap data <i>frame</i>             |
| 9  | <code>FSpectralFlux</code>     | Menghitung nilai fitur spectral flux pada tiap data frame                               |
| 10 | <code>FSpectralRolloff</code>  | Menghitung nilai <i>spectral flux</i> pada tiap data                                    |
| 11 | <code>FZeroCrossing</code>     | Menghitung nilai fitur <i>zero crossing</i> pada tiap data <i>frame</i>                 |
| 12 | <code>getAllFiles</code>       | Mendapatkan lokasi spesifik berkas lagu pada folder terpilih                            |
| 13 | <code>hamming</code>           | Mengembalikan nilai hasil <i>windowing</i> dengan tipe Hamming pada data frame terkait  |
| 14 | <code>length</code>            | Menghitung panjang suatu vektor   |
| 16 | <code>StatisticalValue</code>  | Menghitung nilai rata-rata, standard deviasi, skewness, dan kurtosis pada fitur terkait |

### 3.2.1 Program Utama Ekstraksi Fitur

Program utama adalah program yang digunakan untuk memanggil fungsi-fungsi lain sehingga dapat menjalankan program secara keseluruhan. Pemisahan fungsi-fungsi dari algoritma dilakukan dengan tujuan untuk mempermudah di dalam pengecekan program pada setiap prosesnya. Gambar 3.2.2 hingga

Gambar 3.2.3 menunjukkan *pseudocode* program utama dari tahapan ekstraksi fitur.

|  |   |
|--|---|
| Masukan  | Direktori Tempat Musik<br>disimpan(variebel DIR DATA)         |
| Keluaran   | Nilai fitur pada berkas musik<br>yang diolah (variabel fitur) |
| <pre> 1.  daftarLagu ← getAllFiles(DIR_DATA) 2.  arff←fopen(strcat(DIR_OUTPUT, '\genre music 3.  .arff'), 'w+'); 4.  for i←1 to length(daftarLagu) 5.    [y,Fs]←audioread(daftarLagu(i)) 6.    y ← y(:,1); 7.    frameLength ← floor(0.01*Fs); 8.    curPos ← 1; 9.    step ← floor(0.5*frameLength); 10.   i←1; ← 11.   H ← hamming(frameLength); 12.   while (curPos+frameLength-1)≤length(y)) 13.     x ← y(curPos:curPos+frameLength-1); 14.     frame← H.*(y(curPos:curPos+frameLength-1)); 15.     Mn ← (abs(fft(frame,2*frameLength))); 16.     Mn ← Mn(1:frameLength); 17.     if(i&gt;1) 18.       [SpectralFlux(i-1),Na]← FSpectralFlux(Mn,Nb); 19.       Nb = Na; 20.     else 21.       Nb =Mn/max(Mn); 22.     End 23.     SpectralCentroid(i)← FSpectralCentroid(Mn,Fs,frameLength); 24.     ShortTimeEnergy(i)← FShortTimeEnergy(x); </pre> |   |

**Gambar 3.2.2** Pseudocode program utama (bagian pertama)

```

26.     ZeroCrossing(i) ←
    FZeroCrossing(frame);
27.     SpectralRolloff(i) ←
    FSpectralRolloff(Mn);
28.     SpectralFlatness(i) ←
    FSpectralFlatness(x);
29.     curPos ← curPos + step;
30.     i ← i+1;
31.     end
32.     [fitur(1),fitur(2),fitur(3),fitur(4)] ←
    StatisticalValue(SpectralFlux);
33.     [fitur(5),fitur(6),fitur(7),fitur(8)] ←
    StatisticalValue(SpectralCentroid);
34.     [fitur(9),fitur(10),fitur(11),fitur(12)]
    ← StatisticalValue(ShortTimeEnergy);
35.     [fitur(13),fitur(14),fitur(15),fitur(16)
36.     ] ← StatisticalValue(SpectralRolloff);
37.     [fitur(17),fitur(18),fitur(19),fitur(20)
38.     ] ← StatisticalValue(SpectralFlatness);
39.     C ← strsplit(listSongs{index}, '\ ');
    fprintf(arff, '%s', char(C{length(C)-
40.     1}));
41.     for j=1:length(fitur)
42.         if j~=length(fitur)
43.             fprintf(arff, '%s',
44.                 char(num2str(fitur(j))));
45.         else
46.             fprintf(arff, '%s\n',
47.                 char(num2str(fitur(j))));
48.         end
49.     end
50.     fclose(arff);

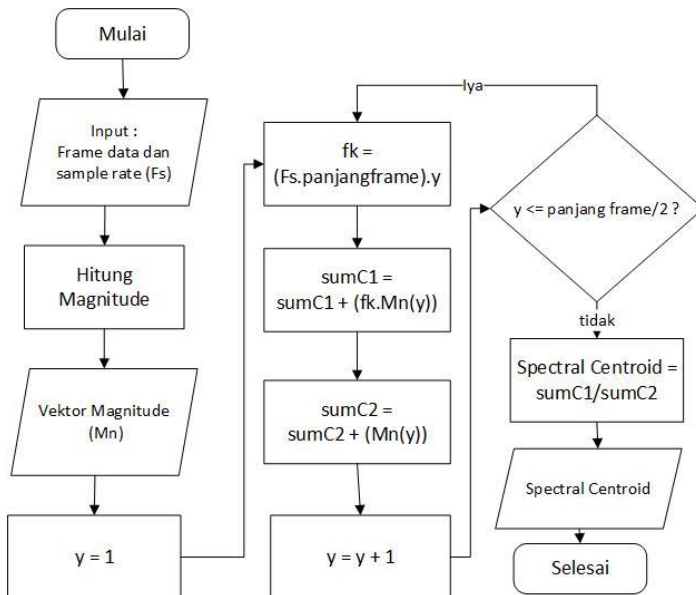
```

**Gambar 3.2.3** Pseudocode program utama (bagian kedua)

### 3.2.2 Spectral Centroid

Program *Spectral centroid* merupakan program yang digunakan untuk mendapatkan nilai fitur *spectral centroid* yang telah dijelaskan pada subbab 2.2.1.

Secara garis besar desain untuk mengekstraksi fitur spectral centroid dapat dilihat pada Gambar 3.2.4. Pada gambar tersebut menjelaskan diagram alir proses ekstraksi fitur spectral centroid. Variabel *sumC1* merupakan variabel yang menyimpan hasil penjumlahan dari perkalian nilai magnitude dengan nilai indeks dari data indeks pertama hingga data indeks terakhir pada *frame*. Dan variabel *sumC2* menyimpan hasil penjumlahan nilai magnitude dari data indeks pertama hingga indeks terakhir pada *frame*. Kemudian *pseudocode* dari program *Spectral Centroid* ditunjukkan pada Gambar 3.2.4 dan Gambar 3.2.5.



**Gambar 3.2.4** Diagram alir proses ekstraksi fitur spectral centroid

|  |  |
|--|--|
| Masukan  | Hasil transformasi fourier data <i>frame</i> (variabel Mn)       |
| Keluaran   | Nilai fitur <i>spectral centroid</i> (variabel SpectralCentroid) |
| <pre> 1.  sumC1 ← 0;sumC2=0; 2.  for j←1tolength(Mn) 3.      sumC1 ← sumC1 + (Mn(j)*j); 4.      sumC2 ← sumC2 + Mn(j); 5.  end 6.  SpectralCentroid ← sumC1/sumC2;</pre> |  |

**Gambar 3.2.5** Pseudocode program ekstraksi fitur *spectral centroid*

### 3.2.3 Spectral Flux

Program *Spectral Flux* merupakan program yang digunakan untuk mendapatkan nilai fitur *spectral flux* yang telah dijelaskan pada subbab 2.2.2.

Secara garis besar desain untuk mengekstraksi fitur spectral centroid dapat dilihat pada gambar 3.2.8. Pada gambar tersebut menjelaskan diagram alir proses ekstraksi fitur spectral flux. *Pseudocode* dari program *Spectral Flux* ditunjukkan pada Gambar 3.2.6 dan 3.2.7.

|          |  |
|----------|--|
| Masukan  | Hasil transformasi fourier data <i>frame</i> (variabel Mn), normalisasi hasil transformasi <i>fourier</i> pada <i>frame</i> sebelumnya (variabel Nb) |
| Keluaran | Nilai fitur <i>spectral flux</i> (variabel SpectralFlux), normalisasi hasil transformasi <i>fourier</i> pada <i>frame</i> sekarang (variabel Na)     |

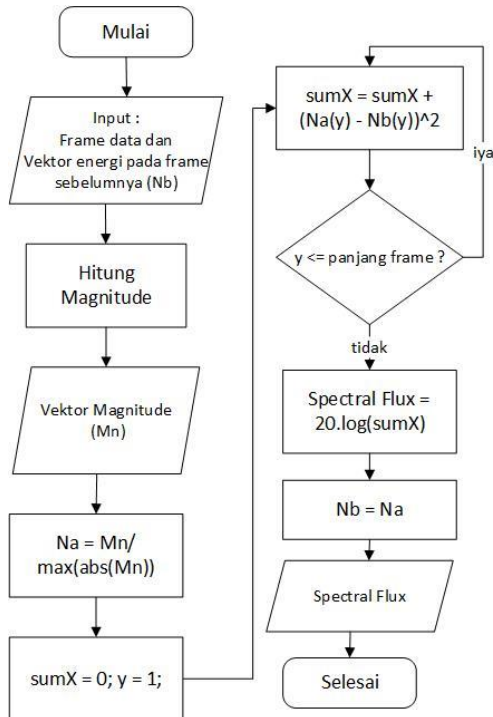
**Gambar 3.2.6** Pseudocode program ekstraksi fitur *spectral flux* (bagian pertama)

```

1.   $N_a \leftarrow M_n / \max(M_n)$  ;
2.  SpectralFlux  $\leftarrow 0$ ;
3.  for  $j \leftarrow 1$  to length( $N_a$ )
4.    SpectralFlux  $\leftarrow$  SpectralFlux + ( $N_a(j) - N_b(j)$ ) . ^2;
5.  end
6.  if (isnan(SpectralFlux) == 1 ||
      isinf(SpectralFlux) == 1)
7.    SpectralFlux  $\leftarrow 0$ ;
8.  end

```

**Gambar 3.2.7** Pseudocode program ekstraksi fitur *spectral flux* (bagian kedua)

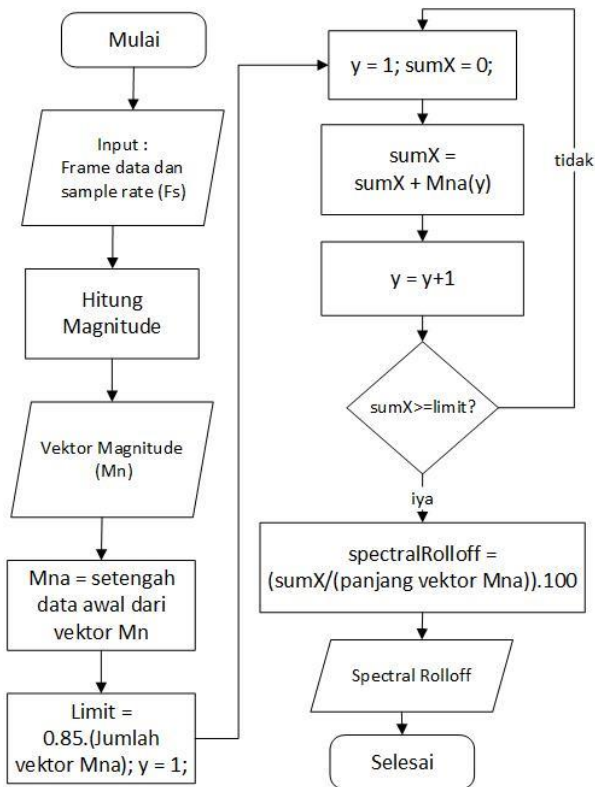


**Gambar 3.2.8** Diagram alir proses *spectral flux*

### 3.2.4 Spectral Rolloff

Program *Spectral Rolloff* merupakan program yang digunakan untuk mendapatkan nilai fitur *spectral rolloff* yang telah dijelaskan pada subbab 2.2.3.

Secara garis besar desain untuk mengekstraksi fitur spectral rolloff dapat dilihat pada gambar 3.2.9. Pada gambar tersebut menjelaskan diagram alir proses ekstraksi fitur spectral rolloff.. *Pseudocode* dari program *Spectral Rolloff* ditunjukkan pada Gambar 3.2.10.



**Gambar 3.2.9** Diagram alur proses ekstraksi fitur spectral rolloff



|   |  |
|---|--|
| Masukan   | Hasil transformasi fourier data <i>frame</i> (variabel Mn) |
| Keluaran  | Nilai fitur <i>spectral rolloff</i> (variabel Rolloff)     |
| 1. Mna $\leftarrow$ Mn;<br>2. limit $\leftarrow$ 0.85*sum(Mna);<br>3. r $\leftarrow$ find(cumsum(Mna(:,1))>= limit,1);<br>4. Rolloff $\leftarrow$ floor((r/(length(Mna)))*100); |  |

**Gambar 3.2.10** Pseudocode program ekstraksi fitur *spectral rolloff*

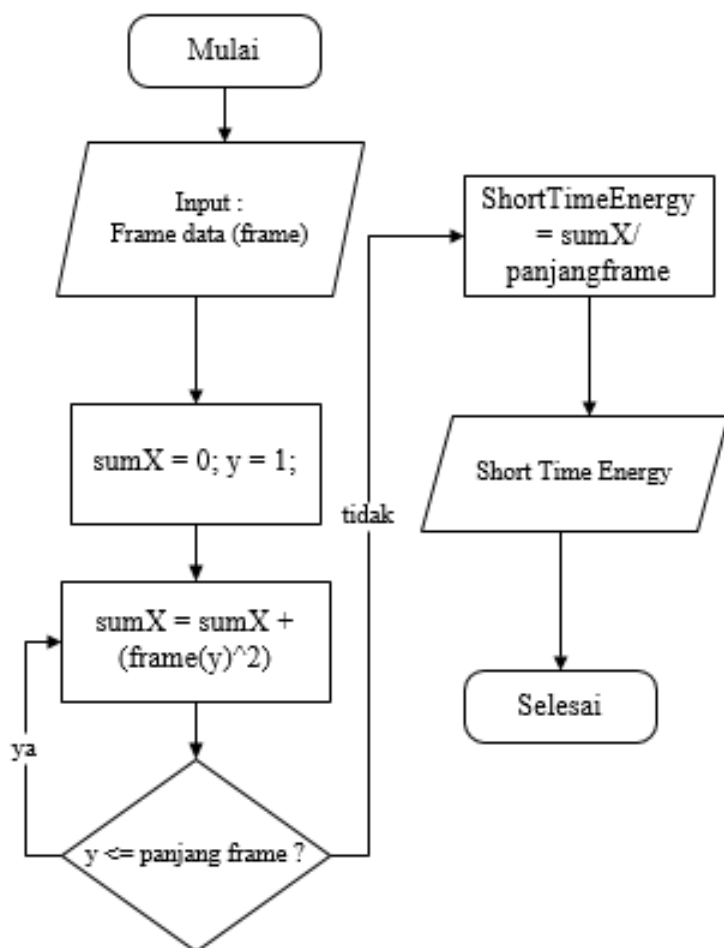
### 3.2.5 Short Time Energy

Program *Short Time Energy* merupakan program yang digunakan untuk mendapatkan nilai fitur *short time energy* yang telah dijelaskan pada subbab 2.2.4.

Secara garis besar desain untuk mengekstraksi fitur spectral rolloff dapat dilihat pada gambar 3.2.12. Pada gambar tersebut menjelaskan diagram alir proses ekstraksi fitur short time energy. Pseudocode dari program *short time energy* ditunjukkan pada Gambar 3.2.11.

|   |  |
|---|--|
| Masukan   | Nilai amplitudo tiap bit pada frame terkait (variabel x) |
| Keluaran  | Nilai fitur <i>short time energy</i> (variabel STE)      |
| 1. H $\leftarrow$ hamming(length(x));<br>2. STE $\leftarrow$ 0;<br>3. for i=1:length(x)<br>4.     STE $\leftarrow$ STE + ((x(i)*H(i))*(x(i)*H(i)));<br>5. end |  |

**Gambar 3.2.11** Pseudocode program ekstraksi fitur *short time energy*

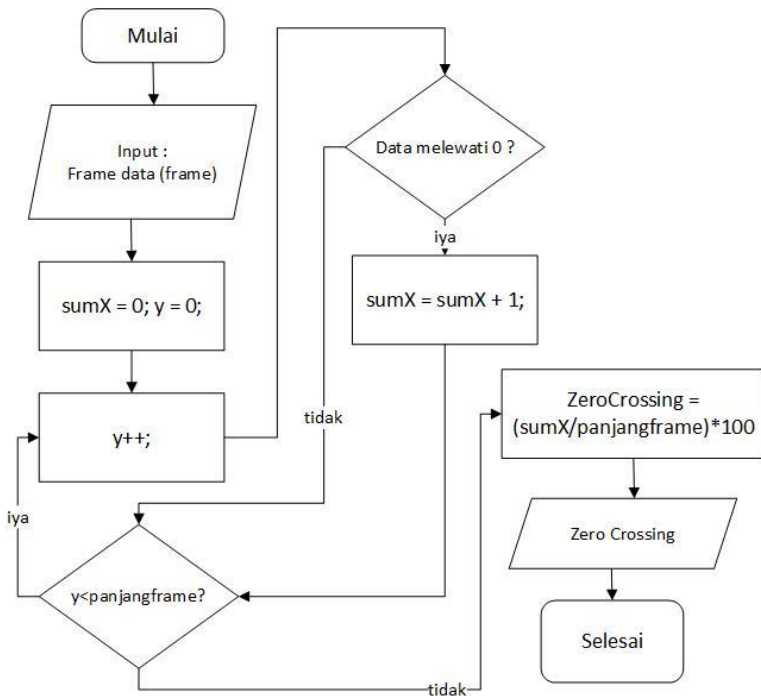


**Gambar 3.2.12** Diagram alir proses ekstraksi fitur short time energy

### 3.2.6 Zero Crossing

Program *Zero Crossing* merupakan program yang digunakan untuk mendapatkan nilai fitur *zero crossing* yang telah dijelaskan pada subbab 2.2.5.

Secara garis besar desain untuk mengekstraksi fitur zero crossing dapat dilihat pada gambar 3.2.13. Pada gambar tersebut menjelaskan diagram alir proses ekstraksi fitur zero crossing. *Pseudocode* dari program *zero crossing* ditunjukkan pada Gambar 3.2.14.



**Gambar 3.2.13** Diagram alir proses ekstraksi fitur zero crossing

|  |  |
|--|--|
| Masukan  | Nilai amplitudo tiap bit pada frame terkait (variabel frame) |
| Keluaran   | Nilai fitur <i>zero crossing</i> (variabel ZeroCrossing)     |
| <pre> 1. Sumx ← 0; 2. for j ← 2 to length(frame) 3.   if (abs(sign(frame(j)) - sign(frame(j-1))) == 2) 4.     Sumx ← Sumx + 1; 5.   end 6. end 7. ZeroCrossing ← (Sumx / length(frame)) * 100;</pre> |  |

**Gambar 3.2.14** *Pseudocode* program ekstraksi fitur *zero crossing*

### 3.2.7 Spectral Flatness

Program ekstraksi fitur spectral flatness merupakan program yang digunakan untuk mendapatkan nilai fitur spectral flatness yang telah dijelaskan pada subbab 2.2.6.

Secara garis besar desain untuk mengekstraksi fitur zero crossing dapat dilihat pada gambar 3.15. Pada gambar tersebut menjelaskan diagram alir proses ekstraksi fitur zero crossing. Pseudocode dari program zero crossing ditunjukkan pada Gambar 3.1.9.

|          |  |
|----------|--|
| Masukan  | Nilai amplitudo tiap bit pada frame terkait (variabel frame)     |
| Keluaran | Nilai fitur <i>spectral flatness</i> (variabel SpectralFlatness) |

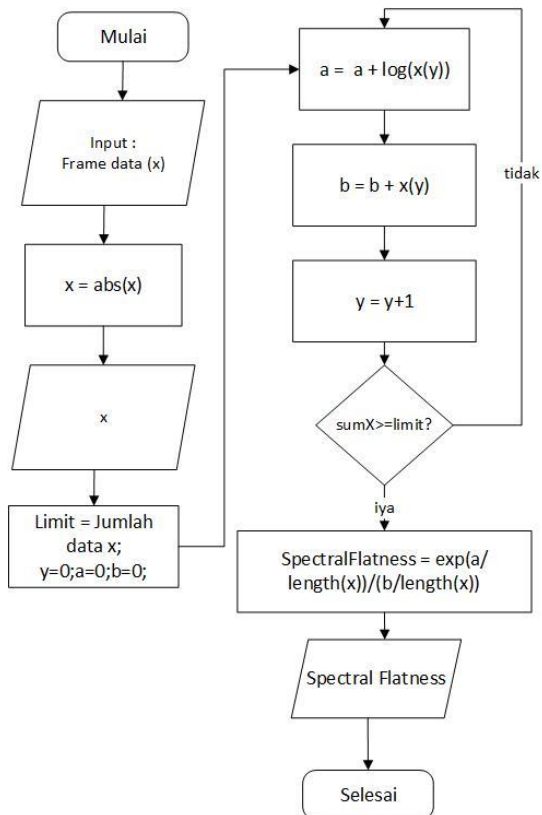
**Gambar 3.2.15** *Pseudocode* program ekstraksi fitur spectral flatness (bagian pertama)

```

1.  frame  $\leftarrow$  abs(frame);
2.  a  $\leftarrow$  0;
3.  b  $\leftarrow$  0;
4.  for j $\leftarrow$ 1 to length(frame)
5.      a  $\leftarrow$  a + log(frame(j));
6.      b  $\leftarrow$  b + frame(j);
7.  end

```

**Gambar 3.2.16** Pseudocode program ekstraksi fitur spectral flatness (bagian kedua)



**Gambar 3.2.17** Diagram alir proses ekstraksi fitur spectral flatness

### 3.3 Perancangan Proses Klasifikasi

Salah satu proses utama pada Tugas Akhir ini yaitu perancangan proses klasifikasi. Pada Tugas Akhir ini, proses klasifikasi untuk menentukan genre pada berkas musik dilakukan menggunakan metode klasifikasi *Random Forest* dan memakai alat bantu Weka versi 3.8. Dataset yang digunakan berformat .ARFF dan merupakan keluaran pada proses ekstraksi fitur yang sebelumnya dilakukan.

Untuk melakukan proses klasifikasi dengan metode *Random Forest* menggunakan alat bantu Weka, dimulai dengan menyiapkan data *training* dan data *testing* terlebih dahulu. Saat dilakukan proses *training*, data *training* yang telah disiapkan digunakan sebagai data sampel. Kemudian dilakukan pengaturan parameter algoritma *Random Forest* yang tersedia pada alat bantu Weka. Parameter tersebut dijelaskan masing-masing kegunaannya pada tabel sekian.

Setelah parameter selesai diatur, barulah algoritma klasifikasi *Random Forest* dijalankan dengan menggunakan data *training* yang telah disiapkan. Alat bantu Weka akan menampilkan hasil trainingnya, kemudian barulah dilanjutkan proses *testing*. Proses *testing* dilakukan dengan menggunakan data *testing* yang telah dipersiapkan sebelumnya sebagai data tes. Selanjutnya algoritma klasifikasi *Random Forest* dijalankan kembali untuk proses *testing*. Alat bantu aplikasi Weka kemudian akan menampilkan hasil *testing* berupa nilai akurasi yang didapatkan dan juga tabel *confussion matrix*.

**Tabel 3.3.1** Daftar parameter pada alat bantu Weka untuk menjalankan proses klasifikasi dengan metode *Random Forest* (bagian pertama)

| No | Parameter      | Keterangan  |
|----|----------------|---|
| 1  | bagSizePercent | Merupakan jumlah data yang digunakan sebagai data <i>training</i> tiap <i>tree</i> dalam satuan presentase dari jumlah data sampel. |

**Tabel 3.3.2** Daftar parameter pada alat bantu Weka untuk menjalankan proses klasifikasi dengan metode *Random Forest* (bagian kedua).

|   |                  |   |
|---|------------------|---|
| 2 | batchSize        | Jumlah data yang digunakan ketika dilakukan <i>batch prediction</i> .               |
| 3 | maxDepth         | Kedalaman maksimal pada <i>tree</i> .   |
| 4 | numDecimalPlaces | Nilai decimal di belakang koma yang digunakan ketika menampilkan hasil perhitungan. |
| 5 | numFeatures      | Jumlah fitur yang digunakan pada tiap <i>tree</i> .                                 |
| 6 | numIterations    | Jumlah iterasi yang dilakukan.  |
| 7 | seed             | Nilai <i>random seed number</i> , yang digunakan untuk menghasilkan angka acak.     |

*[Halaman ini sengaja dikosongkan]*



## **BAB IV IMPLEMENTASI**

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

### **4.1 Lingkungan Implementasi**

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi adalah MATLAB R2015b yang diinstal pada sistem operasi *Windows* 10.

### **4.2 Implementasi**

Pada subbab ini akan dijelaskan implementasi dari setiap subbab yang terdapat pada bab sebelumnya yaitu bab perancangan perangkat lunak. Pada bagian implementasi ini juga akan dibagi menjadi tiga bagian utama, yakni tahapan pra proses, tahapan ekstraksi fitur dan tahapan klasifikasi.

#### **4.2.1 Implementasi Program Utama Tahapan Pra Proses**

Pada bagian ini, dilakukan tahap pra proses. Implementasi tahap ini dilakukan dengan membaca berkas mp3 yang menjadi data yang ingin diambil fiturnya, sehingga didapatkan data sampel dan *sample rate*. Kemudian sampel data dibagi menjadi bagian-bagian dengan ukuran tetap yang disebut dengan frame.

Masukkan pada tahapan pra proses adalah berkas mp3 yang telah memiliki informasi genre. Dan keluaran pada tahapan pra proses ini adalah data frame yang nantinya akan diolah pada tahapan ekstraksi fitur. Implementasi program utama tahapan pra proses ditunjukkan pada pada Kode Sumber 4.1 pada baris 1 hingga baris 13.

#### 4.2.2 Implementasi Program Utama Tahapan Ekstraksi Fitur

Pada bagian ini, dilakukan implementasi tahap ekstraksi fitur. Implementasi tahap ini dilakukan dengan membaca data frame hasil pengolahan pada tahap sebelumnya, yaitu tahapan pra proses. Pada tiap berkas musik diambil *sample data* dan *sample rate* kemudian diambil nilai fitur pada tiap framenya. Selanjutnya nilai fitur tiap frame tersebut dihitung nilai rata-rata, standard deviasi, skewness, dan kurtosisnya. Hasil dari tahap ini adalah dataset berformat .arff yang berisi nilai fitur berkas musik yang telah diekstraksi. Implementasi ini merupakan bagian dari implementasi *pseudocode* pada subbab 3.2.1. Implementasi program utama ekstraksi fitur ditunjukkan pada Kode Sumber 4.1 dan Kode Sumber 4.2.

|    |   |
|----|---|
| 1  | DIR_OUTPUT = 'E:\DataTA\Output';                        |
| 2  | DIR_DATA = 'E:\DataTA\Songs\data ';                     |
| 3  | [listSongs] = getAllFiles(DIR_DATA);                    |
| 4  | arff = fopen(strcat(DIR_OUTPUT, '\output.arff'), 'w+'); |
| 5  | for index=1:length(listSongs)                           |
| 6  | start = cputime;  |
| 7  | [y, Fs] = audioread(listSongs{index});                  |
| 8  | y = y(:,1);   |
| 9  | frameLength = floor(0.01*Fs);                           |
| 10 | curPos = 1;   |
| 11 | step = floor(1*frameLength);                            |
| 12 | i=1;  |
| 13 | H = hamming(frameLength);                               |
| 14 | while (curPos+frameLength-1)<=length(y)                 |
| 15 | x = y(curPos:curPos+frameLength-1);                     |

**Kode Sumber 4.2.1** Implementasi program utama ekstraksi fitur (bagian pertama)

|    |   |
|----|---|
| 16 | frame = H.*(y(curPos:curPos+frameLength-1));                                    |
| 17 | Mn = (abs(fft(frame,2*frameLength)));   |
| 18 | Mn = Mn(1:frameLength);   |
| 19 | if(i>1)   |
| 20 | [SpectralFlux(i-1),Na] = FSpectralFlux(Mn,Nb);                                  |
| 21 | Nb = Na;  |
| 22 | else  |
| 23 | Nb =Mn/max(Mn);   |
| 24 | end   |
| 25 | SpectralCentroid(i) = FSpectralCentroid(Mn,Fs,frameLength);                     |
| 26 | ShortTimeEnergy(i) = FShortTimeEnergy(x);                                       |
| 27 | ZeroCrossing(i) = FZeroCrossing(frame);   |
| 28 | SpectralRolloff(i) = FSpectralRolloff(Mn);                                      |
| 29 | SpectralFlatness(i) = FSpectralFlatness(x);                                     |
| 30 | curPos = curPos + step;   |
| 31 | i=i+1;  |
| 32 | end   |
| 33 | [fitur(1),fitur(2),fitur(3),fitur(4)] = StatisticalValue(SpectralFlux);         |
| 34 | [fitur(5),fitur(6),fitur(7),fitur(8)] = StatisticalValue(SpectralCentroid);     |
| 35 | [fitur(9),fitur(10),fitur(11),fitur(12)] = StatisticalValue(ShortTimeEnergy);   |
| 36 | [fitur(13),fitur(14),fitur(15),fitur(16)] = StatisticalValue(ZeroCrossing);     |
| 37 | [fitur(17),fitur(18),fitur(19),fitur(20)] = StatisticalValue(SpectralRolloff);  |
| 38 | [fitur(21),fitur(22),fitur(23),fitur(24)] = StatisticalValue(SpectralFlatness); |
| 39 | C = strsplit(listSongs{index},'\ ');  |
| 40 | fprintf(arff, '%s,',char(C{length(C)-1}));                                      |
| 41 | var = '';   |

**Kode Sumber 4.2.2** Implementasi program utama ekstraksi fitur  
(bagian kedua)

|    |  |
|----|--|
| 42 | <code>for j=1:length(fitur)</code>                           |
| 43 | <code>if j~=length(fitur)</code>                             |
| 44 | <code>fprintf(arff, '%s, ', char(num2str(fitur(j))));</code> |
| 45 | <code>else</code>  |
| 46 | <code>fprintf(arff, '%s\n', char(num2str(fitur(j))));</code> |
| 47 | <code>end</code>   |
| 48 | <code>end</code>   |
| 49 | <code>end</code>   |
| 50 | <code>fclose(arff);</code>                                   |

**Kode Sumber 4.2.3** Implementasi program utama ekstraksi fitur (bagian ketiga)

#### 4.2.2.1 Ekstraksi Fitur *Spectral Centroid*

Masukan dari program *spectral centroid* berupa hasil transformasi *fourier* sinyal pada frame yang diolah. Data tersebut akan diproses sehingga menghasilkan nilai fitur *spectral centroid*. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.2. Implementasi program ekstraksi fitur *spectral centroid* ditunjukkan pada Kode Sumber 4.2.4

|   |  |
|---|--|
| 1 | <code>function SpectralCentroid = FSpectralCentroid(Mn)</code> |
| 2 | <code>sumC1 = 0; sumC2=0;</code>                               |
| 3 | <code>for j=1:length(Mn)</code>                                |
| 4 | <code>sumC1 = sumC1 + (Mn(j)*j);</code>                        |
| 5 | <code>sumC2 = sumC2 + Mn(j);</code>                            |
| 6 | <code>end</code>   |
| 7 | <code>SpectralCentroid = sumC1/sumC2;</code>                   |

**Kode Sumber 4.2.4** Implementasi fitur *spectral centroid*

#### 4.2.2.2 Ekstraksi Fitur *Spectral Flux*

Masukan dari program *spectral flux* berupa hasil transformasi *fourier* sinyal pada frame yang diolah dan hasil

normalisasi transformasi *fourier* pada frame sebelumnya. Data tersebut akan diproses sehingga menghasilkan nilai fitur *spectral flux*. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.3. Implementasi program ekstraksi fitur spectral flux ditunjukkan pada Kode Sumber 4.2.5

|  |                            |   |                |
|--|----------------------------|---|----------------|
|  | <code>function</code>      | <code>[SpectralFlux,Na]</code>                    | <code>=</code> |
|  | <code>FSpectralFlux</code> | <code>(Mn,Nb)</code>                              |                |
|  | <code>Na</code>            | <code>= Mn/max(Mn);</code>                        |                |
|  | <code>SpectralFlux</code>  | <code>= 0;</code>                                 |                |
|  | <code>for</code>           | <code>j=1:length(Na)</code>                       |                |
|  | <code>SpectralFlux</code>  | <code>= SpectralFlux + (Na(j) - Nb(j)).^2;</code> |                |
|  | <code>end</code>           |   |                |

**Kode Sumber 4.2.5** Implementasi fitur *spectral flux*

#### 4.2.2.3 Ekstraksi Fitur *Spectral Rolloff*

Masukan dari program *spectral rolloff* berupa hasil transformasi *fourier* sinyal pada frame yang diolah. Data tersebut akan diproses sehingga menghasilkan nilai fitur *spectral rolloff*. Implementasi ini juga merupakan implementasi *pseudocode* pada subbab 3.2.4. Implementasi program ekstraksi fitur spectral rolloff ditunjukkan pada Kode Sumber 4.2.6.

|   |                               |  |                |
|---|-------------------------------|--|----------------|
| 1 | <code>function</code>         | <code>SpectralRolloff</code>                         | <code>=</code> |
|   | <code>FSpectralRolloff</code> | <code>(Mn)</code>                                    |                |
| 2 | <code>Mna</code>              | <code>= Mn</code>                                    |                |
| 3 | <code>limit</code>            | <code>= 0.85*sum(Mna);</code>                        |                |
| 4 | <code>r</code>                | <code>= find(cumsum(Mna(:,1)) &gt;= limit,1);</code> |                |
| 5 | <code>SpectralRolloff</code>  | <code>= floor((r/(length(Mna)))*100);</code>         |                |

**Kode Sumber 4.2.6** Implementasi fitur *spectral rolloff*

#### 4.2.2.4 Ekstraksi Fitur *Short Time Energy*

Masukan dari program *short time energy* berupa data sinyal pada frame yang diolah. Data tersebut akan diproses sehingga menghasilkan nilai fitur *short time energy*. Implementasi ini merupakan implementasi *pseudocode* pada subbab 4.2.7. Implementasi program fungsi ekstraksi fitur short time energy ditunjukkan pada Kode Sumber 4.2.7.

|   |   |
|---|---|
| 1 | <code>function STE = FShortTimeEnergy(x)</code>     |
| 2 | <code>H = hamming(length(x));</code>                |
| 3 | <code>STE = 0;</code>                               |
| 4 | <code>for i=1:length(x)</code>                      |
| 5 | <code>STE = STE + ((x(i)*H(i))*(x(i)*H(i)));</code> |
| 6 | <code>end</code>                                    |

**Kode Sumber 4.2.7** Implementasi fitur *short time energy*

#### 4.2.2.5 Ekstraksi Fitur *Zero Crossing*

Masukan dari program ekstraksi fitur *zero crossing* berupa data sinyal pada frame yang diolah. Data tersebut akan diproses sehingga menghasilkan nilai fitur *zero crossing*. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.6. Implementasi program untuk fungsi proses ekstraksi fitur *zero crossing* ditunjukkan pada Kode Sumber 4.2.8.

|   |   |
|---|---|
| 1 | <code>function ZeroCrossing = FZeroCrossing(x)</code> |
| 2 | <code>Sumx = 0;</code>                                |
| 3 | <code>for j=2:length(x)</code>                        |
| 4 | <code>if(abs(sign(x(j))-sign(x(j-1))))==2)</code>     |
| 5 | <code>Sumx = Sumx+1;</code>                           |
| 6 | <code>end</code>                                      |
| 7 | <code>end</code>                                      |
| 8 | <code>ZeroCrossing = (Sumx/length(x))*100;</code>     |

**Kode Sumber 4.2.8** Implementasi fitur *zero crossing*

### 4.2.3 Ekstraksi Fitur Spectral Flatness

Masukan dari program ekstraksi fitur *zero crossing* berupa data sinyal pada frame yang diolah. Data tersebut akan diproses sehingga menghasilkan nilai fitur *zero crossing*. Implementasi ini merupakan implementasi *pseudocode* pada subbab 3.2.6. Implementasi program untuk fungsi proses ekstraksi fitur *zero crossing* ditunjukkan pada Kode Sumber 4.2.8.

|   |  |   |
|---|--|---|
| 1 | <code>function SpectralFlatness</code>       | = |
|   | <code>FSpectralFlatness(x);</code>           |   |
| 2 | <code>a=0;b=0;</code>                        |   |
| 3 | <code>for j=1:length(x)</code>               |   |
| 4 | <code>    a = a+log(x(j));</code>            |   |
| 5 | <code>    b = b+x(j);</code>                 |   |
| 6 | <code>end</code>                             |   |
| 7 | <code>SpectralFlatness</code>                | = |
|   | <code>exp(a/length(x))/(b/length(x));</code> |   |

**Kode Sumber 4.2.9** Implementasi ekstraksi fitur spectral flatness

#### 4.2.3.1 Penghitungan Nilai Statistik

Masukan dari program perhitungan nilai statistik berupa data hasil ekstraksi fitur pada tiap frame yang dibentuk. Keluaran dari proses ini adalah nilai fitur yang nantinya akan dipakai pada tahapan proses klasifikasi. Implementasi program ini merupakan implementasi *pseudocode* yang telah dijabarkan pada subbab sekian. Implementasi program untuk fungsi proses penghitungan nilai statistik ditunjukkan pada Kode Sumber sekian.

|   |  |   |
|---|--|---|
| 1 | <code>function [mean,std,skewness,kurtosis]</code>         | = |
|   | <code>StatisticalValue(data)</code>                        |   |
| 2 | <code>mean = sum(data)/length(data);</code>                |   |
| 3 | <code>std=sqrt(sum((data-mean).^2)/(length(data)));</code> |   |

**Kode Sumber 4.2.10** Implementasi fungsi penghitungan nilai statistik (bagian pertama)

|   |   |
|---|---|
| 4 | <code>skewness=sum((data-mean).^3)/(length(data)-1)*(std.^3);</code>                                |
| 5 | <code>kurtosis = ((sum((data-mean).^4)/length(data))/(sum((data-mean).^2)/length(data)).^2);</code> |

**Kode Sumber 4.2.11** Implementasi fungsi penghitungan nilai statistik (bagian kedua)

#### 4.2.4 Implementasi Klasifikasi

Pada bagian ini, dilakukan implementasi tahap klasifikasi menggunakan metode random forest dengan alat bantu aplikasi Weka. Implementasi tahap ini dilakukan dengan membaca data training dan testing berformat .ARFF hasil keluaran pada tahapan proses sebelumnya yaitu proses ekstraksi fitur.



## BAB V

### UJI COBA DAN EVALUASI

Bab ini merupakan bahasan mengenai hasil uji coba dan evaluasi terhadap hasil proses ekstraksi fitur musik untuk mengelompokkan berkas musik berdasarkan genre. Pembahasan pada bab ini meliputi lingkungan uji coba, data uji coba, serta skenario uji coba dan evaluasi dari setiap pengujian.

#### 5.1 Lingkungan Pelaksanaan Uji Coba

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi algoritma ekstraksi fitur musik berdasarkan genre pada Tugas Akhir ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam uji coba aplikasi dapat dilihat pada table 5.1.

**Tabel 5.1.1** Lingkungan Uji Coba

|                 |                      |                                   |
|-----------------|----------------------|-----------------------------------|
| Perangkat Keras | Prosesor             | Intel® Core™ i5 CPU @ 2.40GHz     |
|                 | Memori RAM           | 8 GB                              |
| Perangkat Lunak | Sistem Operasi       | Microsoft Windows 10 Pro 64-bit   |
|                 | Perangkat Pengembang | MATLAB 8.6.0 (R2015b)<br>Weka 3.8 |

#### 5.2 Data Uji Coba

Data masukan direpresentasikan dalam file dengan format .arff. Data uji coba yang digunakan pada tahapan uji coba adalah berupa 1000 buah file musik dengan 10 genre yaitu genre musik blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, dan rock yang diunduh dari website <http://freemusicarchive.org>. Data yang digunakan berupa berkas musik berformat .mp3.

Data pengujian dibagi menjadi dua yaitu data untuk *training* dan data untuk *testing*. Terdapat 70 file musik tiap genre, dengan total 700 data untuk training dan 30 file musik tiap genre, dengan total 300 data untuk testing.

### 5.3 Uji Coba Ekstraksi Fitur

Pada subbab ini dijelaskan mengenai uji coba hasil tahapan proses ekstraksi fitur, untuk mengetahui apakah fitur yang diekstraksi telah menjadi karakteristik pembeda antara berkas musik yang memiliki genre berbeda. Hal ini akan dijelaskan dalam bentuk grafik hasil ekstraksi fitur tiap nilai fitur yang dilakukan pada frame-frame yang telah dibentuk.

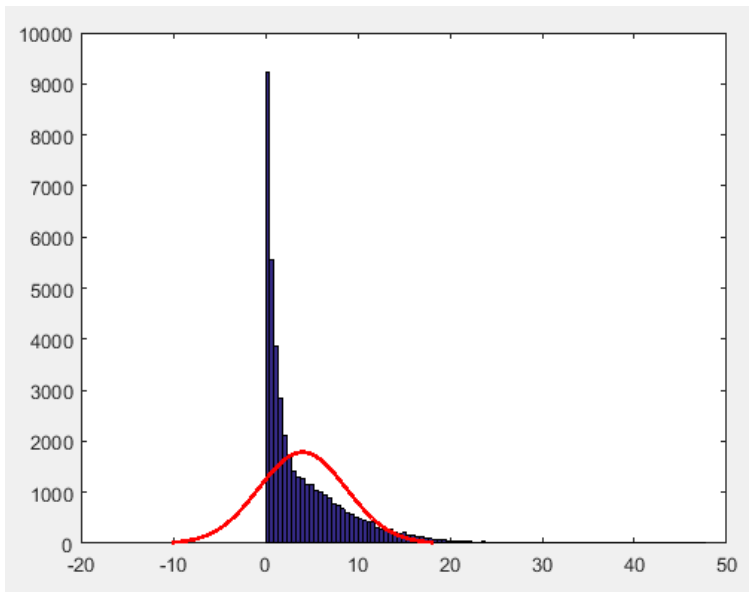
Subbab 5.3.1 hingga subbab 5.3.6 menunjukkan histogram data nilai fitur pada tiap-tiap frame yang terbentuk. Pada histogram tersebut, sumbu x menunjukkan nilai fitur yang dihasilkan dan sumbu y menunjukkan frekuensi data. Garis lengkung merah pada histogram tersebut menunjukkan bentuk distribusi data hasil perhitungan nilai fitur dalam satu *frame*.

Uji coba ekstraksi fitur dilakukan dengan menghitung nilai fitur dari data 11 berkas mp3, untuk kemudian dianalisis nilainya sehingga dapat menghasilkan kesimpulan mengenai karakteristik yang ada pada musik yang direpresentasikan oleh nilai hasil ekstraksi fitur pada hasil uji coba yang dilakukan.

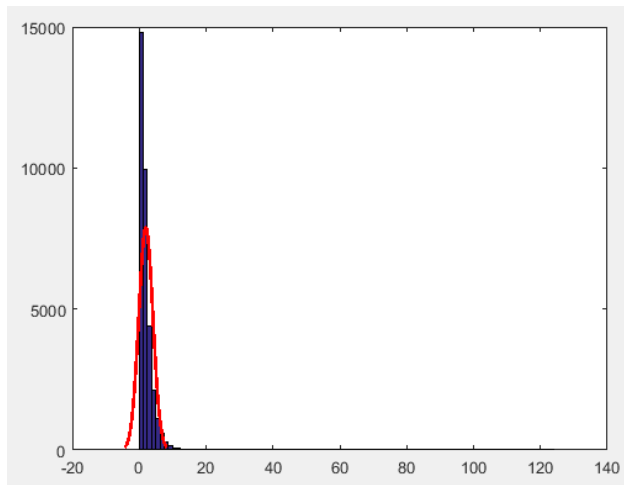
Tujuan dilakukannya uji coba ekstraksi fitur pada subbab ini adalah untuk mengetahui pengaruh nilai dari masing-masing fitur yang diolah terhadap karakteristik yang melekat pada berkas mp3 yang diuji dan juga mengetahui bagaimana perbedaan distribusi data hasil perhitungan nilai fitur terhadap 11 berkas mp3 dengan genre yang berbeda. Untuk mencapai tujuan tersebut, maka hasil dari perhitungan nilai fitur pada tiap frame yang dihasilkan direpresentasikan dalam bentuk histogram untuk masing-masing berkas mp3 yang diuji. Dan juga hasil perhitungan nilai fitur, ditunjukkan dalam bentuk tabel, sehingga dapat dianalisa keterkaitan nilai fitur yang didapat dengan karakteristik yang melekat pada berkas mp3 tersebut.

### 5.3.1 Uji Coba Ekstraksi Fitur Spectral Flux

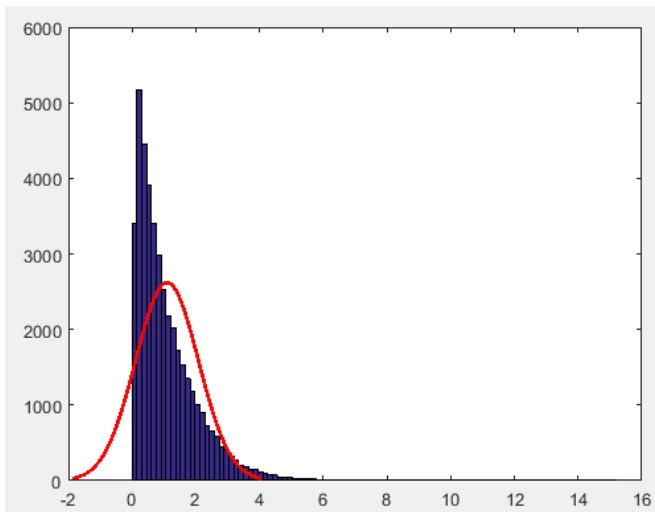
Pada subbab ini, dijelaskan mengenai hasil uji coba perbandingan bentuk histogram data nilai fitur di tiap frame-frame yang terbentuk dengan jenis berkas musik yang berbeda genre atau karakteristik. Perbandingan histogram antar genre pada fitur spectral flux ditunjukkan pada Gambar 5.3.1 hingga Gambar 5.3.11.



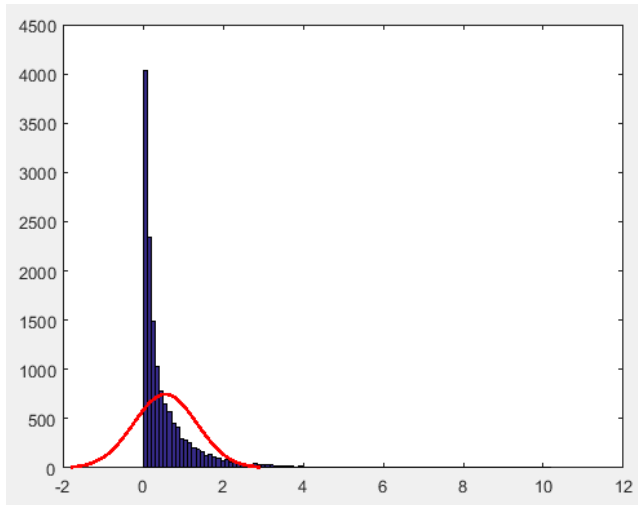
**Gambar 5.3.1** Histogram nilai fitur spectral flux pada berkas mp3 buku audio



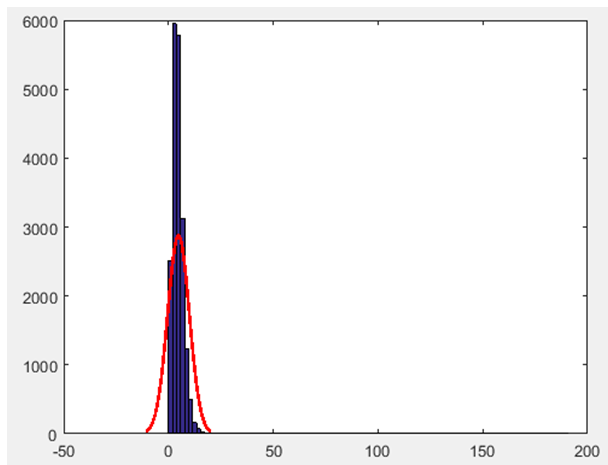
**Gambar 5.3.2** Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre metal



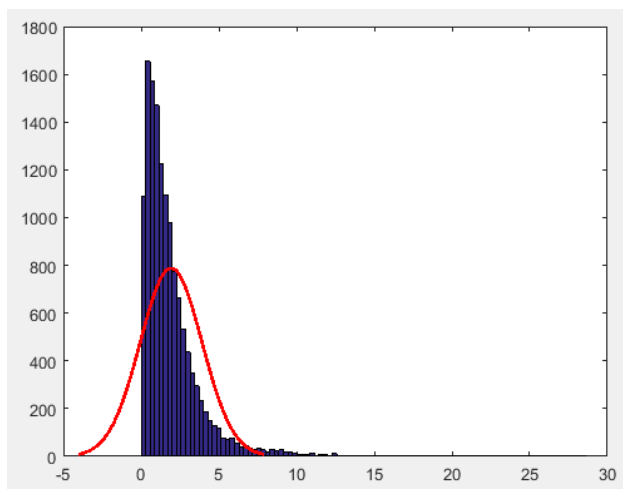
**Gambar 5.3.3** Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre blues



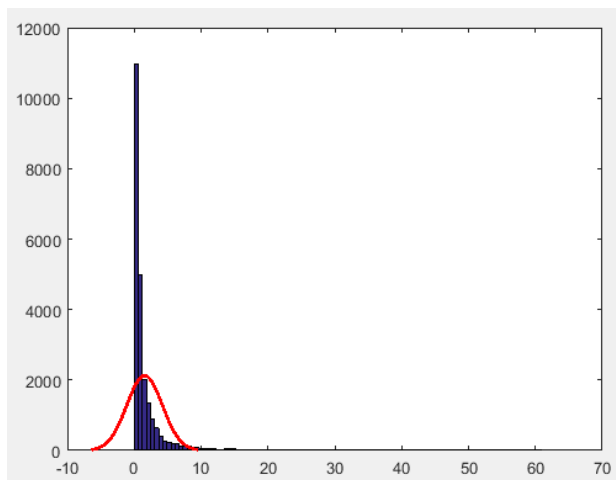
**Gambar 5.3.4** Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre classical



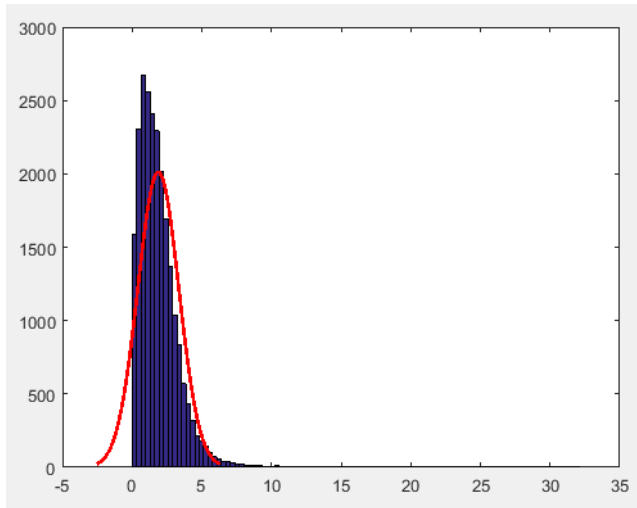
**Gambar 5.3.5** Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre country



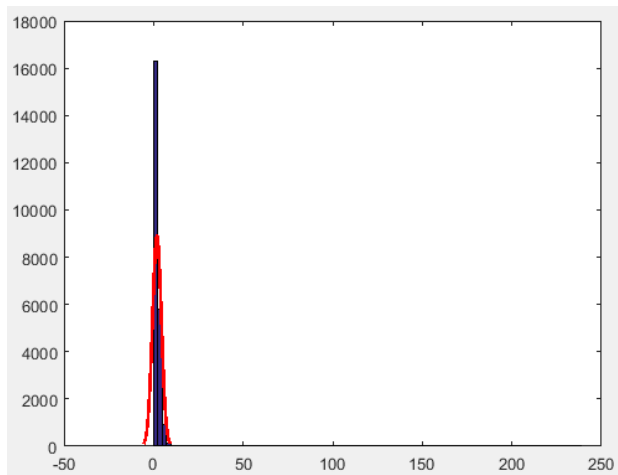
**Gambar 5.3.6** Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre disco



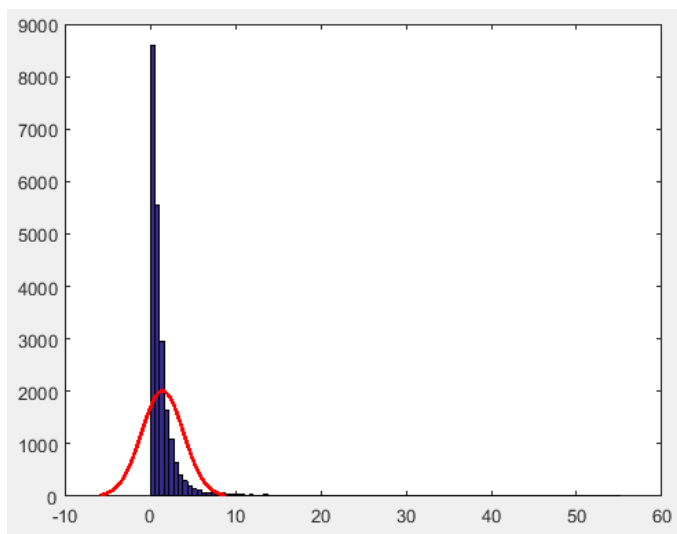
**Gambar 5.3.7** Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre hip hop



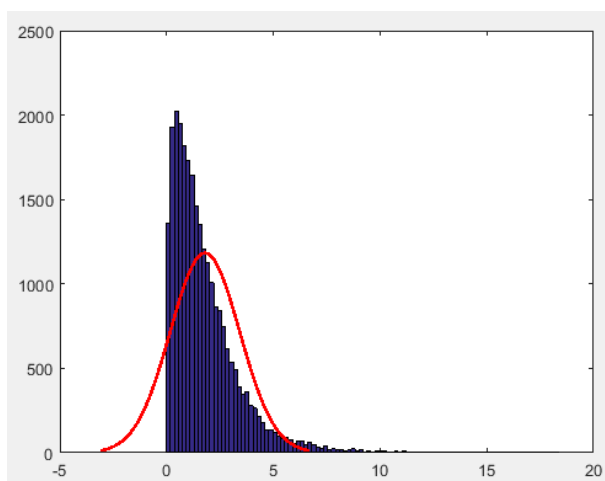
**Gambar 5.3.8** Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre jazz



**Gambar 5.3.9** Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre pop



**Gambar 5.3.10** Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre reggae



**Gambar 5.3.11** Histogram nilai fitur spectral flux pada berkas mp3 musik dengan genre rock



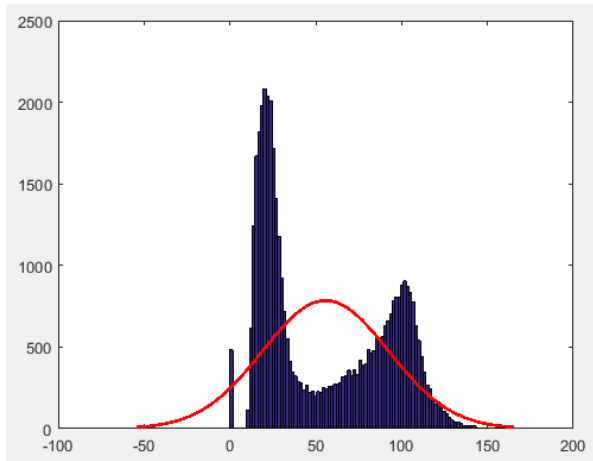
**Tabel 5.3.1** Hasil perhitungan nilai statistik data nilai fitur spectral flux pada frame-frame yang terbentuk

| Jenis berkas | Mean | Standard Deviasi | Skewness | Kurtosis |
|--------------|------|------------------|----------|----------|
| Pop          | 1.9  | 2.48             | 46.45    | 3989.17  |
| Metal        | 1.98 | 2.11             | 12.86    | 629.47   |
| Country      | 4.77 | 5.16             | 16.02    | 403.86   |
| Reggae       | 1.46 | 2.45             | 5.73     | 54.15    |
| HipHop       | 1.54 | 2.66             | 4.46     | 36.24    |
| Jazz         | 1.88 | 1.47             | 3.04     | 30.07    |
| Disco        | 1.93 | 1.99             | 3.27     | 21.47    |
| Classical    | 0.56 | 0.79             | 3.1      | 17.16    |
| Rock         | 1.8  | 1.63             | 2.27     | 11.82    |
| Blues        | 1.09 | 0.99             | 1.98     | 10.61    |
| Audio book   | 4.02 | 4.71             | 1.88     | 7.7      |

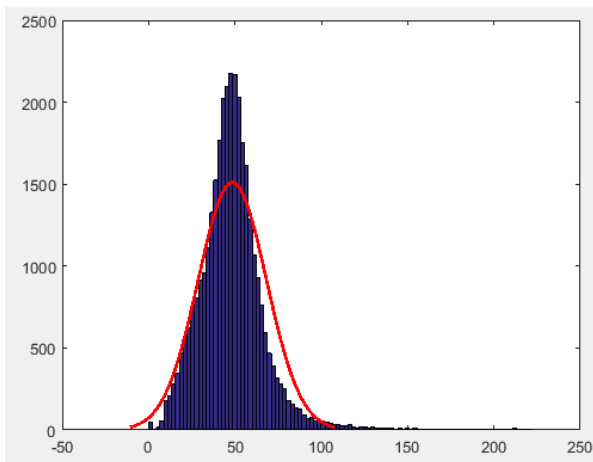
Dari data Tabel 5.3.1 dan dapat disimpulkan bahwa nilai kurtosis pada data nilai fitur spectral flux akan semakin besar apabila semakin ramai atau bergejolak alunan nada pada lagu yang diolah.

### 5.3.2 Uji Coba Ekstraksi Fitur Spectral Centroid

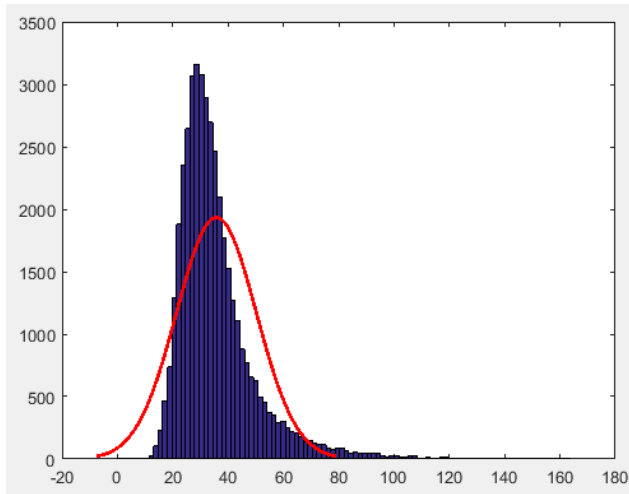
Pada subbab ini, dijelaskan mengenai hasil uji coba perbandingan bentuk histogram data nilai fitur di tiap frame-frame yang terbentuk dengan jenis berkas musik yang berbeda genre atau karakteristik. Perbandingan histogram antar genre pada fitur spectral centroid ditunjukkan pada Gambar 5.3.12 hingga Gambar 5.3.22. Dan perbandingan hasil penghitungan nilai statistik pada data fitur spectral centroid, dijelaskan pada Tabel 5.3.2.



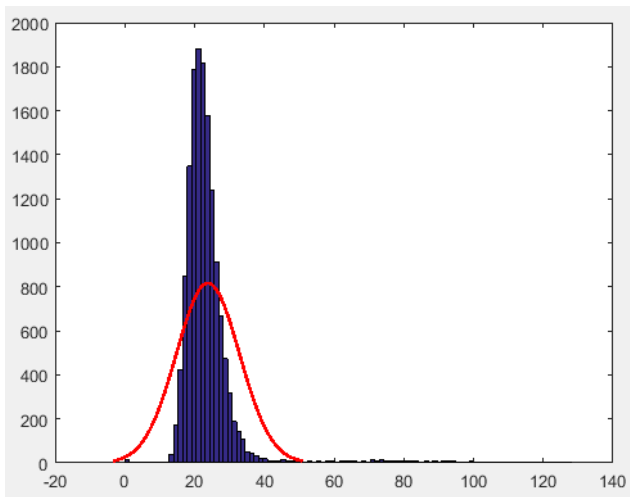
**Gambar 5.3.12** Histogram nilai fitur spectral centroid pada berkas mp3 buku audio



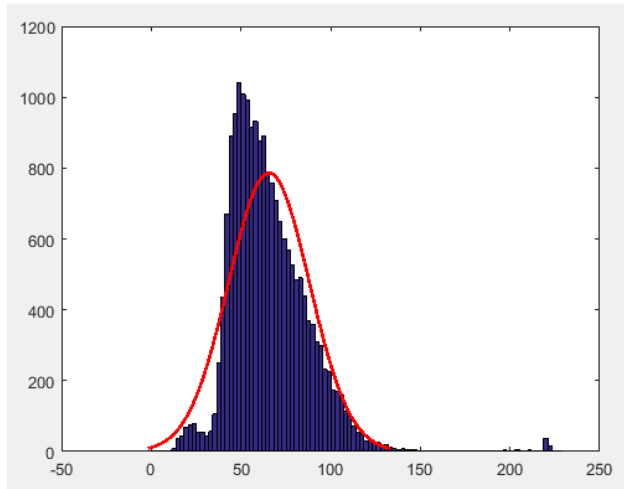
**Gambar 5.3.13** Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre metal



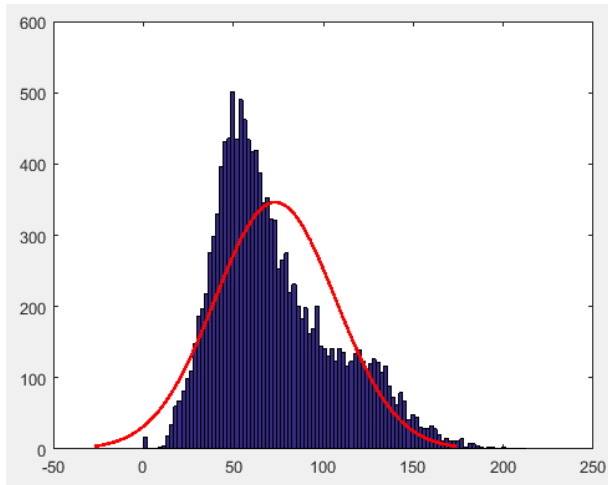
**Gambar 5.3.14** Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre blues



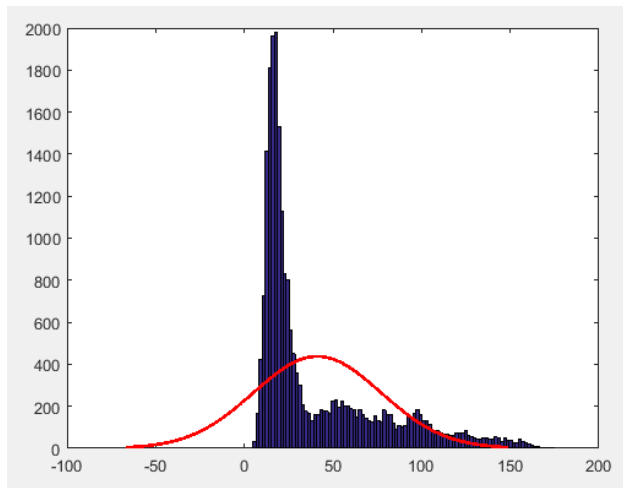
**Gambar 5.3.15** Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre classical



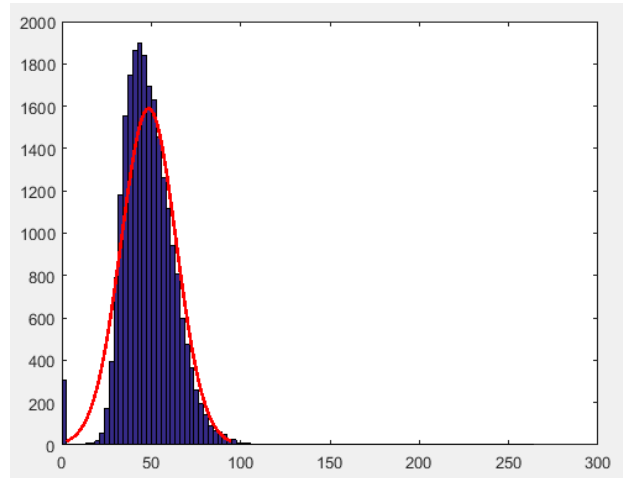
**Gambar 5.3.16** Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre country



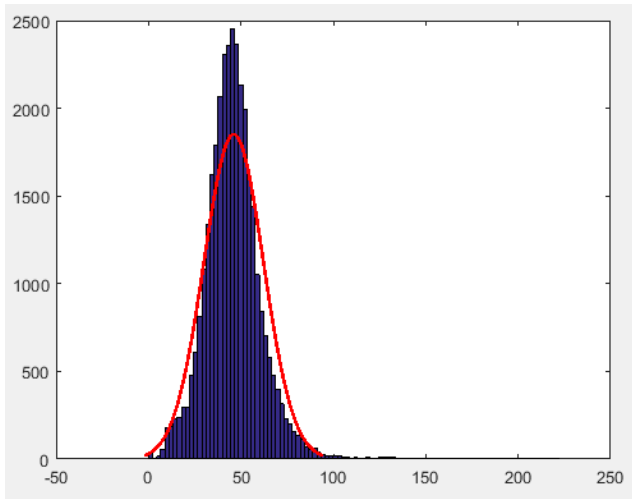
**Gambar 5.3.17** Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre disco



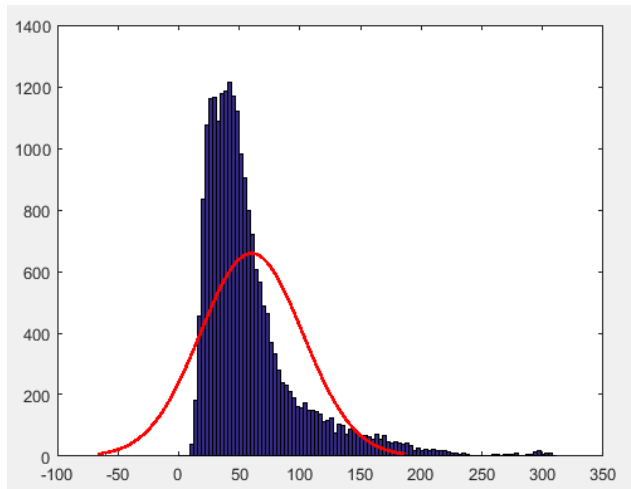
**Gambar 5.3.18** Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre hip hop



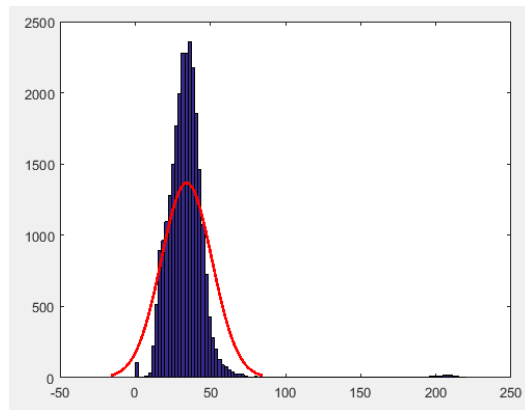
**Gambar 5.3.19** Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre jazz



**Gambar 5.3.20** Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre pop



**Gambar 5.3.21** Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre reggae



**Gambar 5.3.22** Histogram nilai fitur spectral centroid pada berkas mp3 musik dengan genre rock

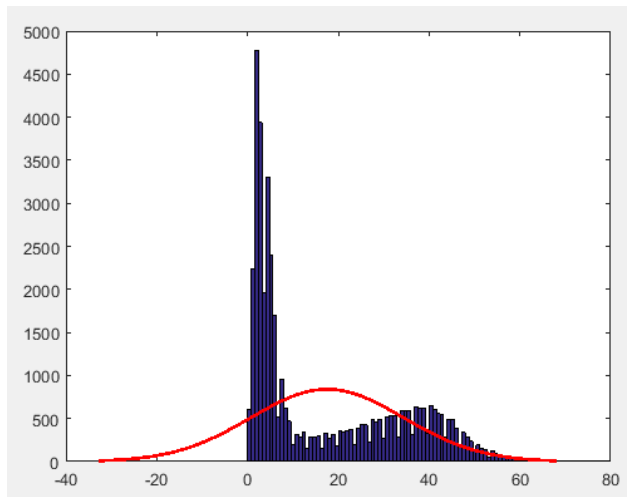
**Tabel 5.3.2** Hasil perhitungan nilai statistik data nilai fitur spectral centroid pada frame-frame yang terbentuk

| Jenis berkas | Mean  | Standard Deviasi | Skewness | Kurtosis |
|--------------|-------|------------------|----------|----------|
| Disco        | 73.34 | 35.51            | 0.79     | 3.08     |
| Country      | 65.63 | 22.53            | 1.72     | 11.25    |
| Reggae       | 60.1  | 42.17            | 2.2      | 9.34     |
| Audio book   | 55.73 | 36.65            | 0.34     | 1.57     |
| Jazz         | 48.55 | 15.36            | 1.95     | 27.31    |
| Metal        | 48.49 | 19.75            | 1.9      | 13.44    |
| Pop          | 46.08 | 16.11            | 1.78     | 15.07    |
| HipHop       | 41.15 | 35.91            | 141      | 4.04     |
| Blues        | 35.8  | 14.42            | 2.18     | 10.61    |
| Rock         | 34.25 | 16.83            | 6        | 59.76    |
| Classical    | 23.82 | 9.06             | 5.07     | 36.64    |

Data hasil penghitungan nilai statistik fitur spectral centroid pada tiap berkas mp3 dengan genre yang dibandingkan ditunjukkan pada Tabel 5.3.2. Dari data Tabel 5.3.2 dan dapat disimpulkan bahwa semakin terang alunan nada suatu lagu, maka nilai *mean* dan *standard deviasi* pada fitur spectral centroid semakin besar.

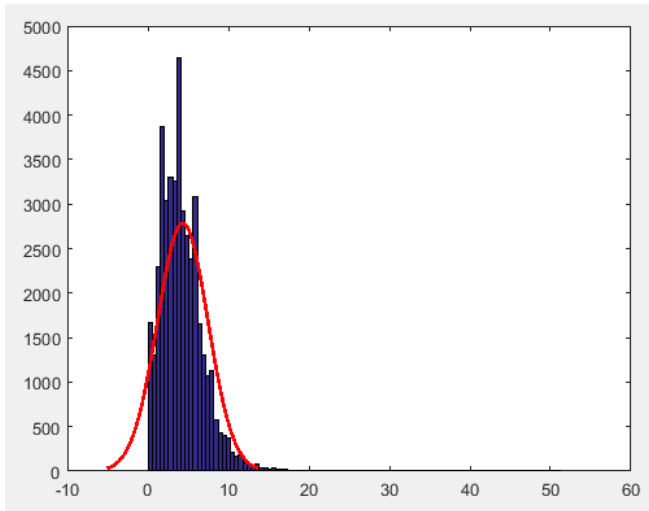
### 5.3.3 Uji Coba Ekstraksi Fitur Zero Crossing

Pada subbab ini, dijelaskan mengenai hasil uji coba perbandingan bentuk histogram data nilai fitur zero crossing di tiap frame-frame yang terbentuk dengan jenis berkas musik yang berbeda genre atau karakteristik. Perbandingan histogram antar genre pada fitur zero crossing ditunjukkan pada Gambar 5.3.231 hingga Gambar 5.3.33. Dan perbandingan hasil penghitungan nilai statistik pada data fitur zero crossing, dijelaskan pada Tabel 5.3.3.

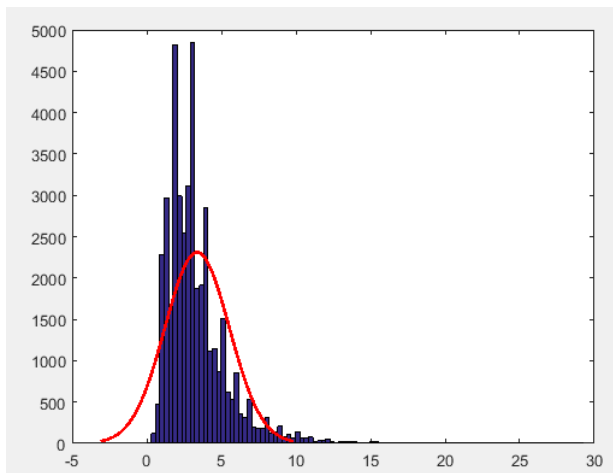


**Gambar 5.3.23** Histogram nilai fitur zero crossing pada berkas mp3 buku audio

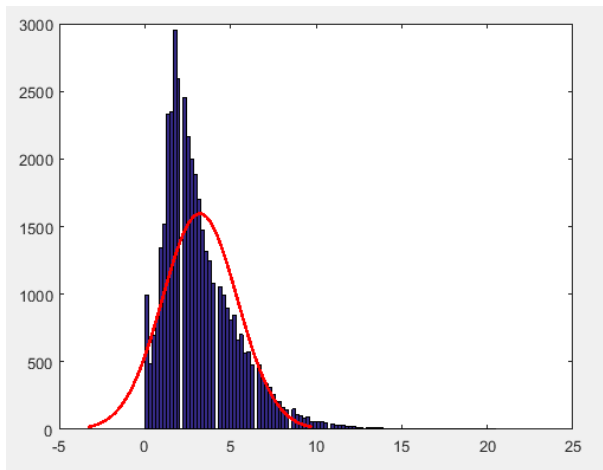




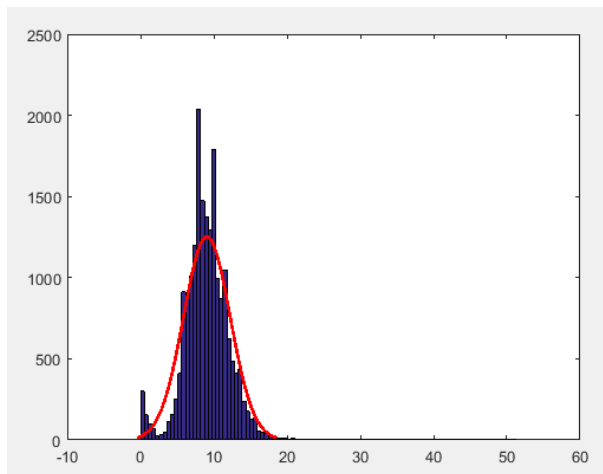
**Gambar 5.3.24** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre metal



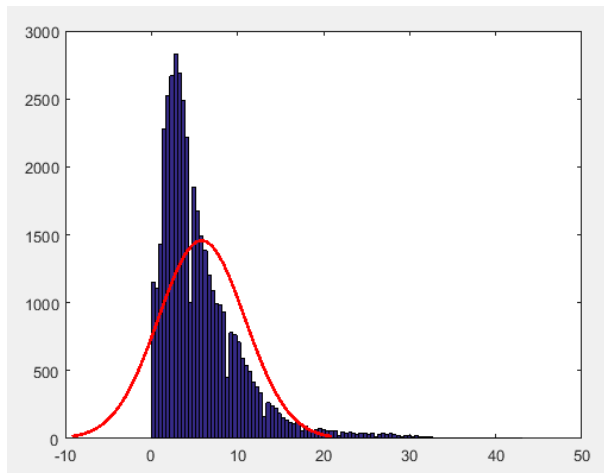
**Gambar 5.3.25** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre blues



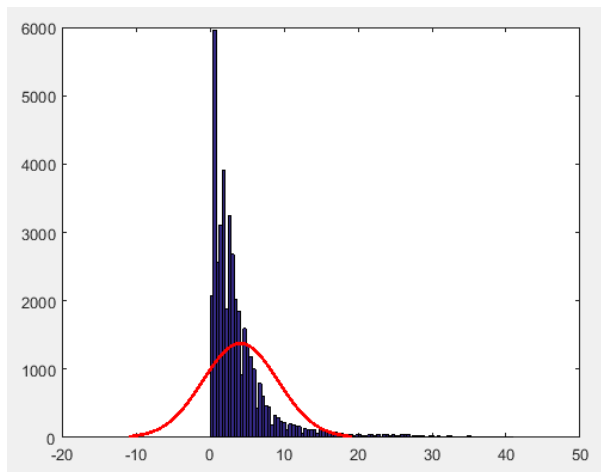
**Gambar 5.3.26** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre classical



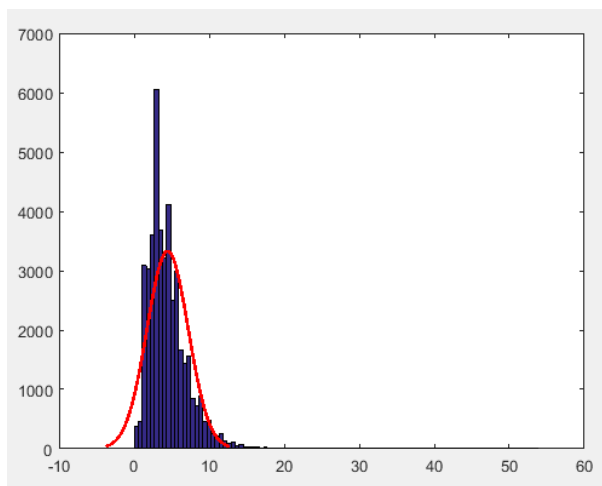
**Gambar 5.3.27** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre country



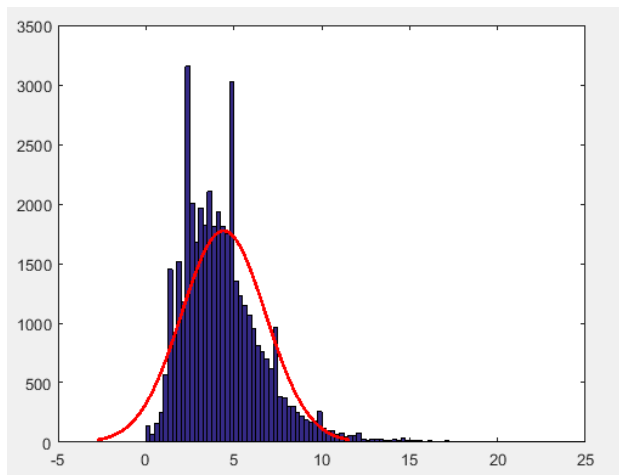
**Gambar 5.3.28** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre disco



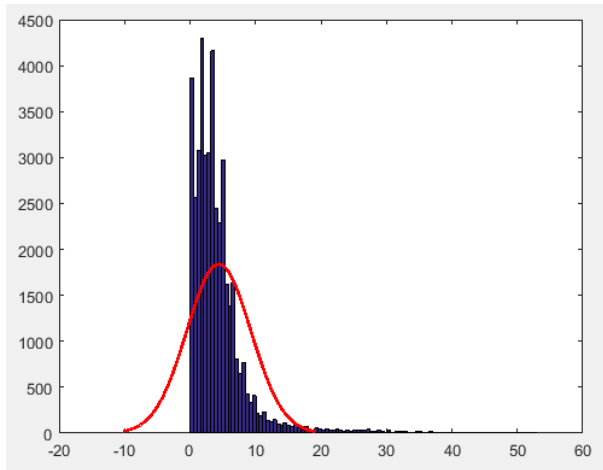
**Gambar 5.3.29** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre hip hop



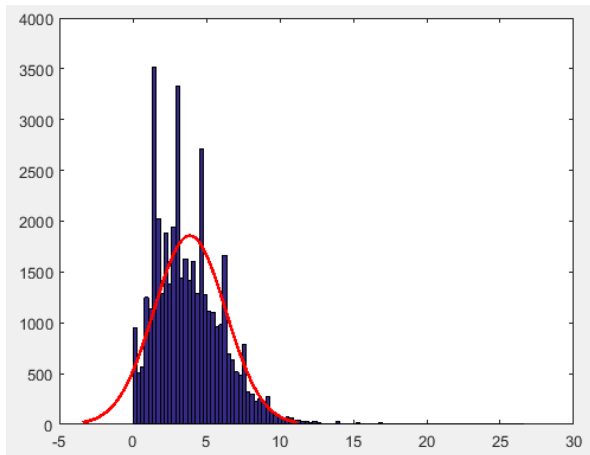
**Gambar 5.3.30** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre jazz



**Gambar 5.3.31** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre pop



**Gambar 5.3.32** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre reggae



**Gambar 5.3.33** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre rock

Data hasil penghitungan nilai statistik fitur zero crossing pada tiap berkas mp3 dengan genre yang dibandingkan ditunjukkan pada Tabel 5.3.3. Dari data Tabel 5.3.3 dan dapat disimpulkan bahwa semakin banyak loncatan alunan nada (dentuman/*beat*) pada lagu maka nilai *mean* dari data nilai fitur zero crossing akan semakin besar.

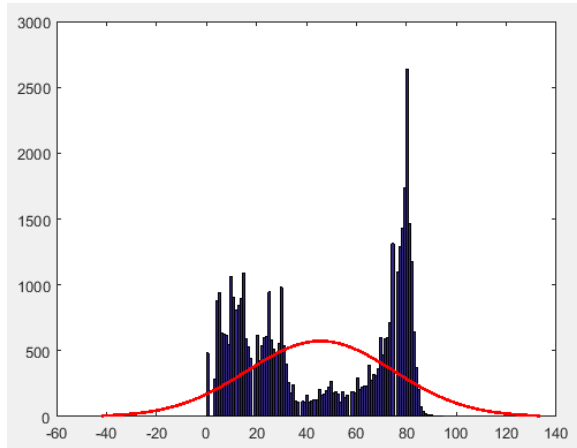
**Tabel 5.3.3** Hasil perhitungan nilai statistik data nilai fitur zero crossing pada frame-frame yang terbentuk

| Jenis berkas | Mean  | Standard Deviasi | Skewness | Kurtosis |
|--------------|-------|------------------|----------|----------|
| Audio book   | 17.55 | 16.82            | 0.72     | 2.14     |
| Metal        | 4.6   | 3.32             | 4.24     | 46.08    |
| Blues        | 3.38  | 2.16             | 2.4      | 13.48    |
| Classical    | 2.02  | 1.04             | 0.98     | 5.71     |
| Country      | 9.05  | 3.18             | 0.99     | 14.72    |
| Disco        | 8.27  | 6.88             | 1.52     | 5.49     |
| HipHop       | 4.42  | 6.48             | 2.58     | 9.97     |
| Jazz         | 5.25  | 2.93             | 2.98     | 34.53    |
| Pop          | 5.15  | 2.32             | 1.43     | 7.65     |
| Reggae       | 5.02  | 6.26             | 3.09     | 14.85    |
| Rock         | 4.09  | 2.48             | 1.4      | 8.22     |

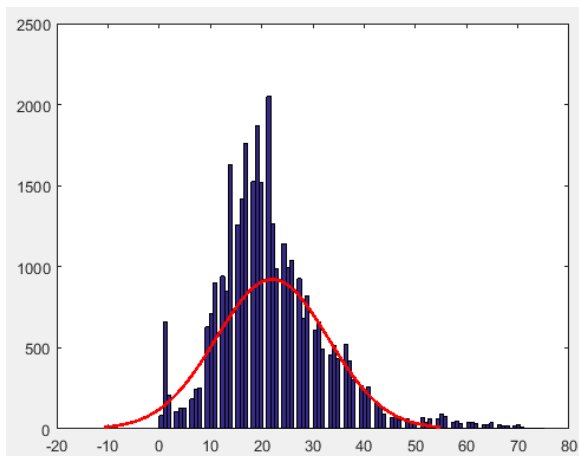
### 5.3.4 Uji Coba Ekstraksi Fitur Spectral Rolloff

Pada subbab ini, dijelaskan mengenai hasil uji coba perbandingan bentuk histogram data nilai fitur spectral rolloff di tiap frame-frame yang terbentuk dengan jenis berkas musik yang berbeda genre atau karakteristik. Perbandingan histogram antar genre pada fitur spectral rolloff ditunjukkan pada Gambar 5.3.34

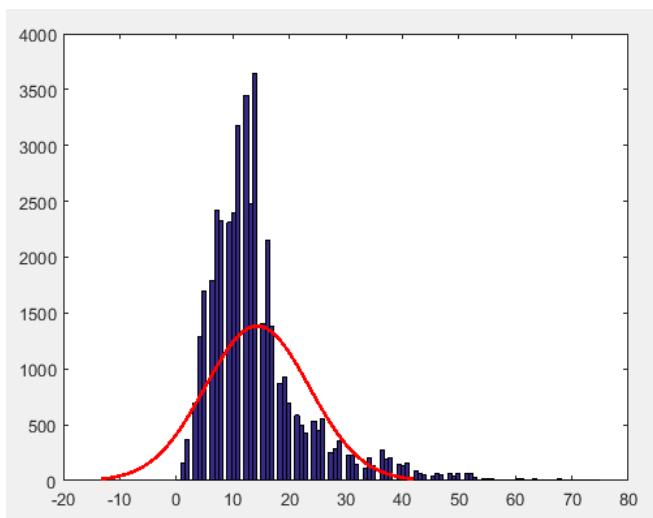
hingga Gambar 5.3.44. Dan perbandingan hasil penghitungan nilai statistik pada data fitur zero crossing, dijelaskan pada Tabel 5.3.4.



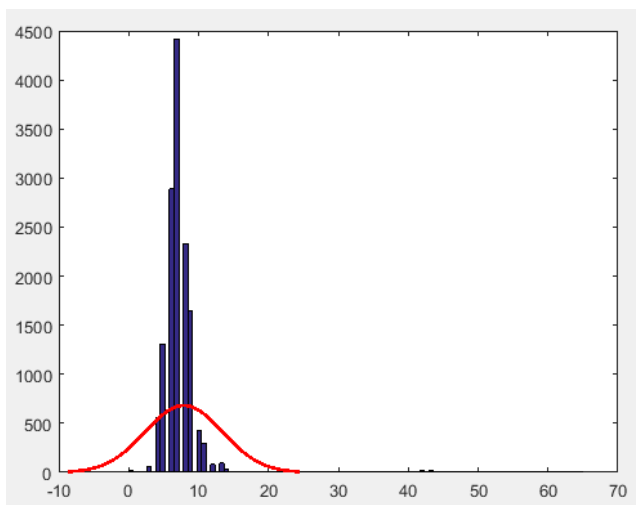
**Gambar 5.3.34** Histogram nilai fitur zero crossing pada berkas mp3 buku audio



**Gambar 5.3.35** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre metal

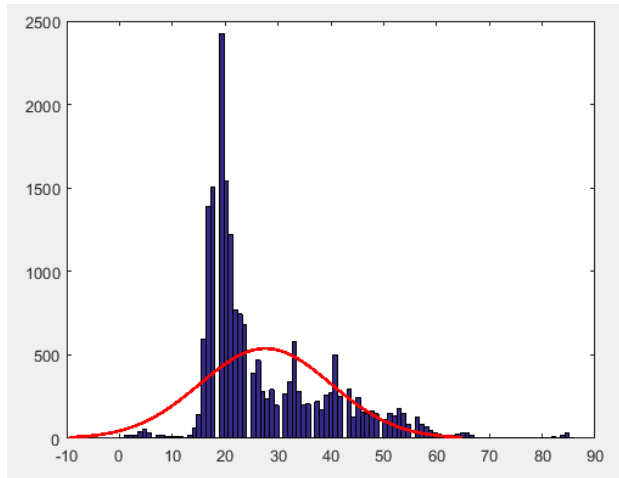


**Gambar 5.3.36** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre blues

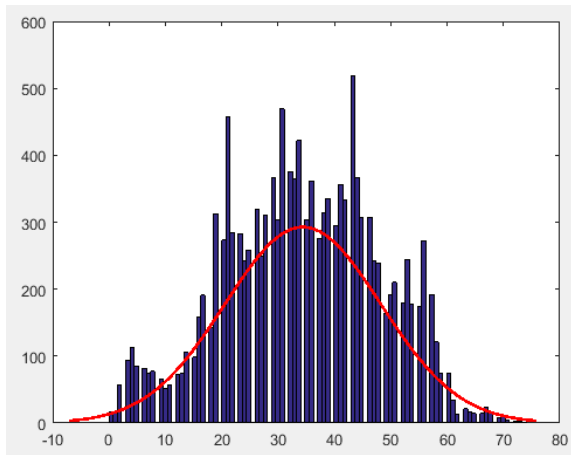


**Gambar 5.3.37** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre classical

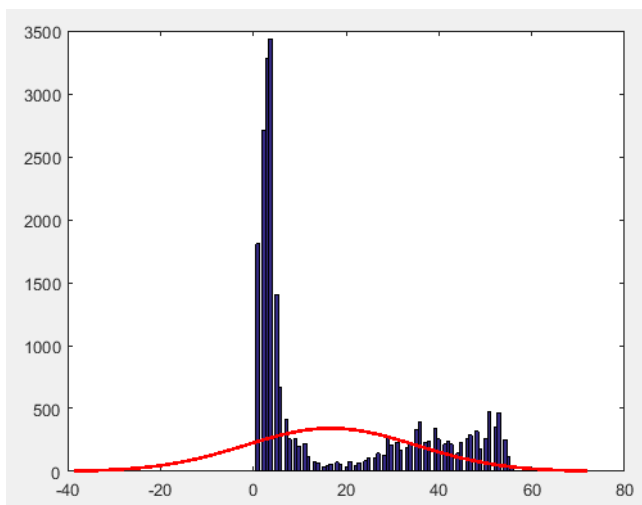




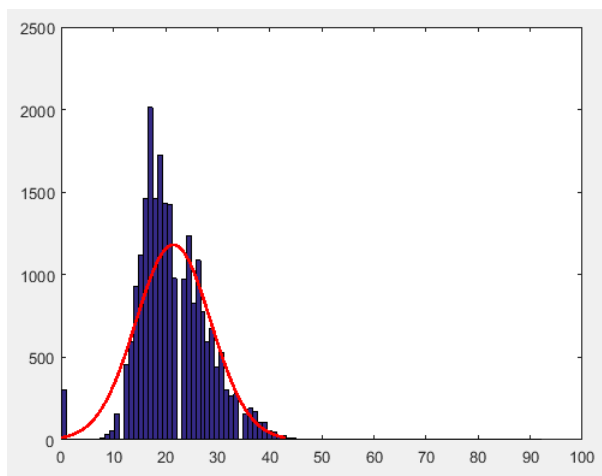
**Gambar 5.3.38** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre country



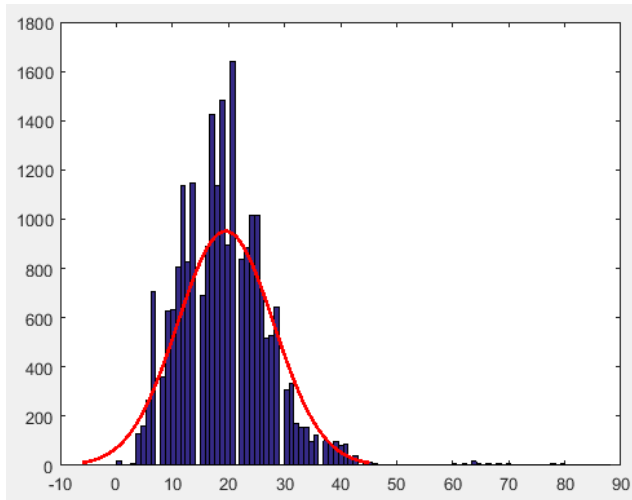
**Gambar 5.3.39** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre disco



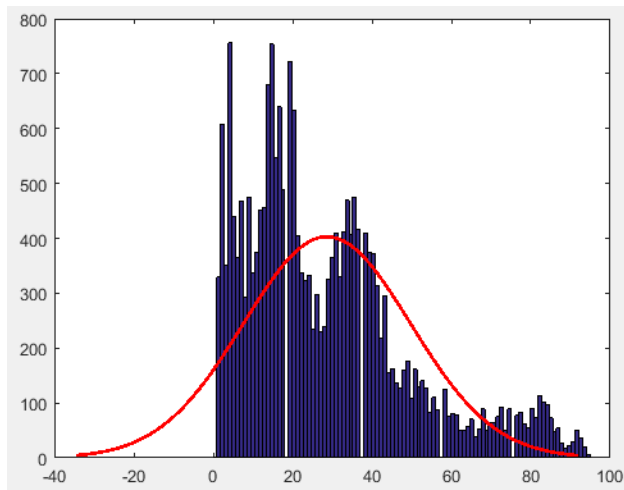
**Gambar 5.3.40** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre hip hop



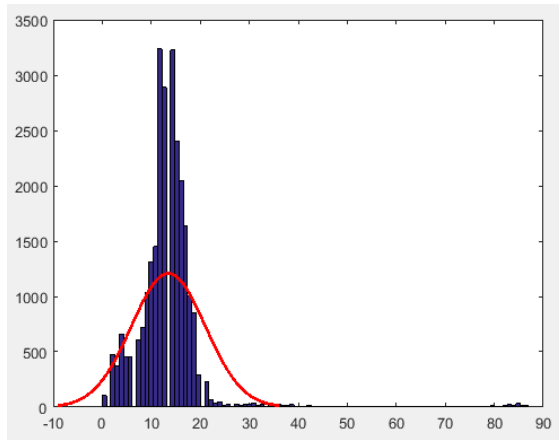
**Gambar 5.3.41** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre jazz



**Gambar 5.3.42** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre pop



**Gambar 5.3.43** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre reggae



**Gambar 5.3.44** Histogram nilai fitur zero crossing pada berkas mp3 musik dengan genre rock

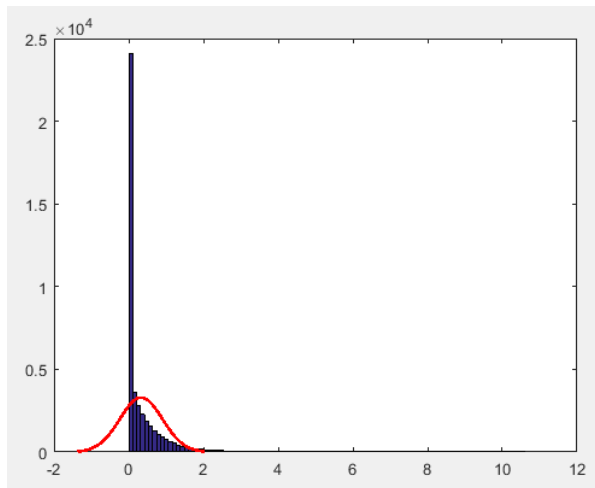
**Tabel 5.3.4** Hasil perhitungan nilai statistik data nilai fitur spectral rolloff pada frame-frame yang terbentuk

| Jenis berkas | Mean  | Standard Deviasi | Skewness | Kurtosis |
|--------------|-------|------------------|----------|----------|
| Audio book   | 45.58 | 29.26            | -0.03    | 1.33     |
| Disco        | 34.4  | 13.84            | -0.13    | 2.53     |
| Reggae       | 28.66 | 21.07            | 1        | 3.51     |
| Country      | 27.62 | 12.4             | 1.28     | 4.58     |
| Metal        | 22.05 | 10.93            | 1.05     | 5.22     |
| Jazz         | 21.49 | 7.18             | 1.2      | 11.89    |
| Pop          | 19.51 | 8.56             | 1.56     | 10.36    |
| HipHop       | 16.72 | 18.45            | 0.85     | 2.06     |
| Blues        | 14.32 | 9.22             | 1.92     | 8.01     |
| Rock         | 13.49 | 7.53             | 5.25     | 47.62    |
| Classical    | 7.88  | 5.51             | 6.36     | 47.84    |

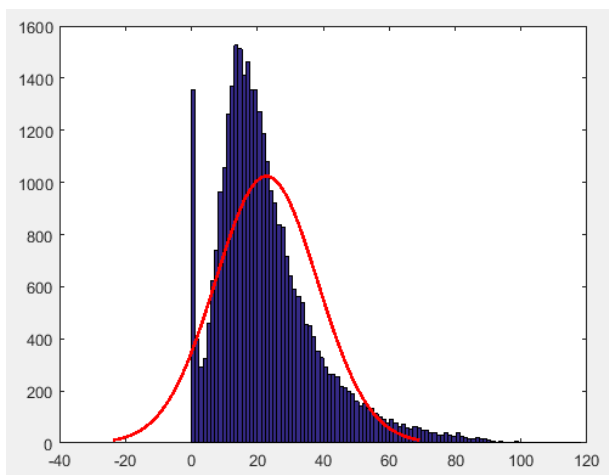
Data hasil penghitungan nilai statistik fitur spectral rolloff pada tiap berkas mp3 dengan genre yang dibandingkan ditunjukkan pada Tabel 5.3.4. Dari data Tabel 5.3.4 dapat disimpulkan bahwa semakin besar kerapatan suara pada lagu maka nilai *mean* data nilai fitur spectral rolloff akan semakin besar dan nilai kurtosisnya semakin kecil.

### 5.3.5 Uji Coba Ekstraksi Fitur Short Time Energy

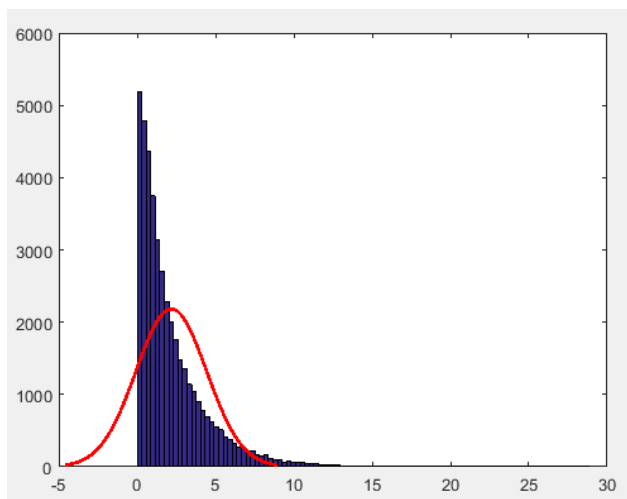
Pada subbab ini, dijelaskan mengenai hasil uji coba perbandingan bentuk histogram data nilai fitur short time energy di tiap frame-frame yang terbentuk dengan jenis berkas musik yang berbeda genre atau karakteristik. Perbandingan histogram antar genre pada fitur short time energy ditunjukkan pada Gambar 5.3.45 hingga Gambar 5.3.55. Dan perbandingan hasil penghitungan nilai statistik pada data fitur short time energy, dijelaskan pada Tabel 5.3.5.



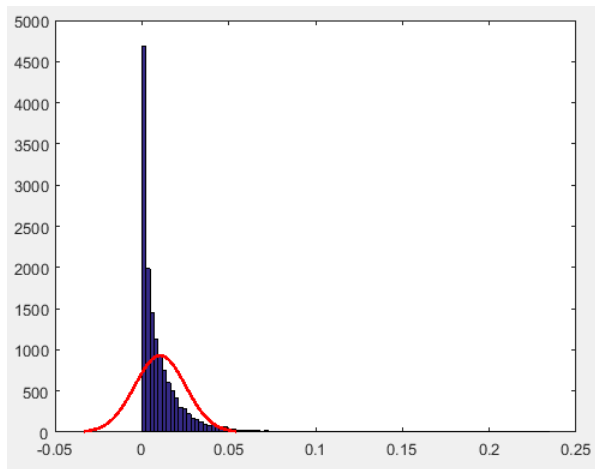
**Gambar 5.3.45** Histogram nilai fitur short time energy pada berkas mp3 buku audio



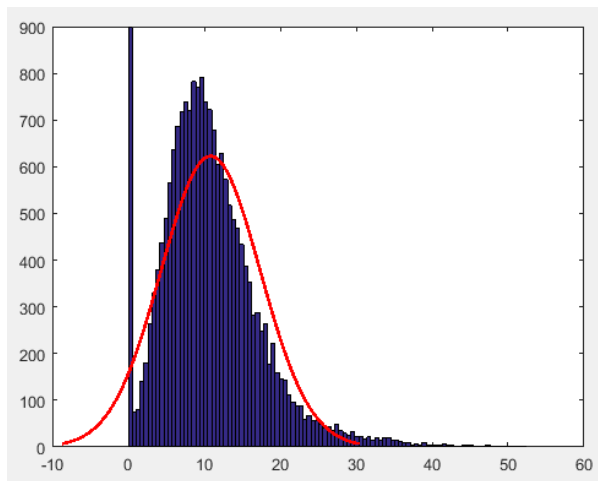
**Gambar 5.3.46** Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre metal



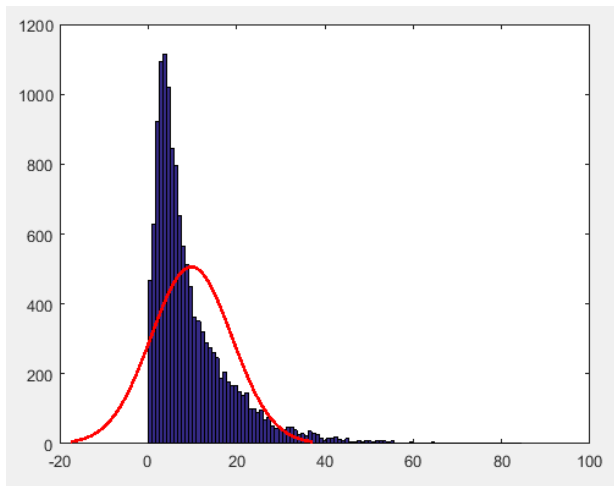
**Gambar 5.3.47** Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre blues



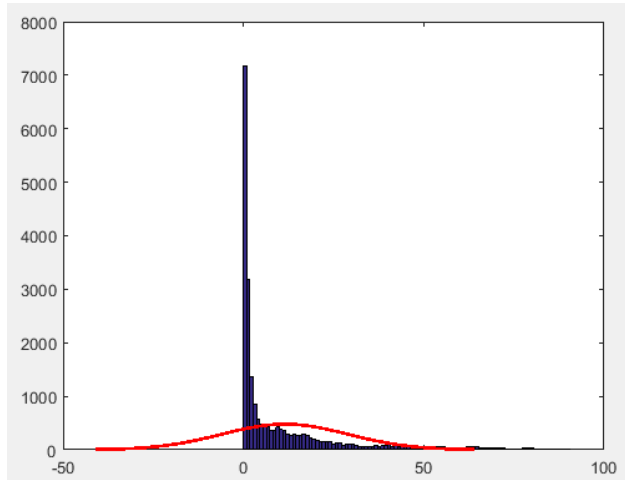
**Gambar 5.3.48** Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre classical



**Gambar 5.3.49** Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre country

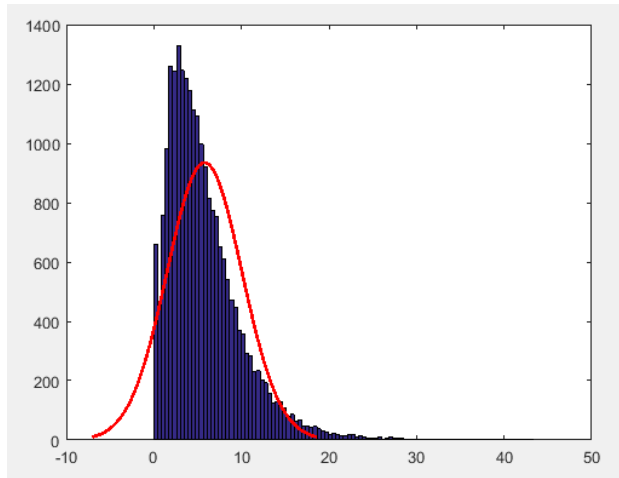


**Gambar 5.3.50** Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre disco

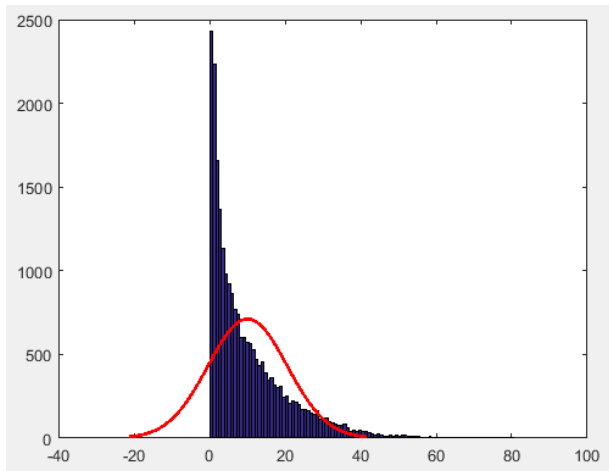


**Gambar 5.3.51** Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre hip hop

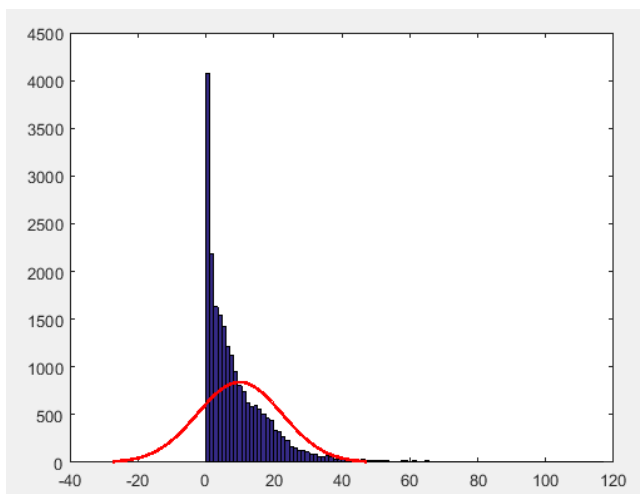




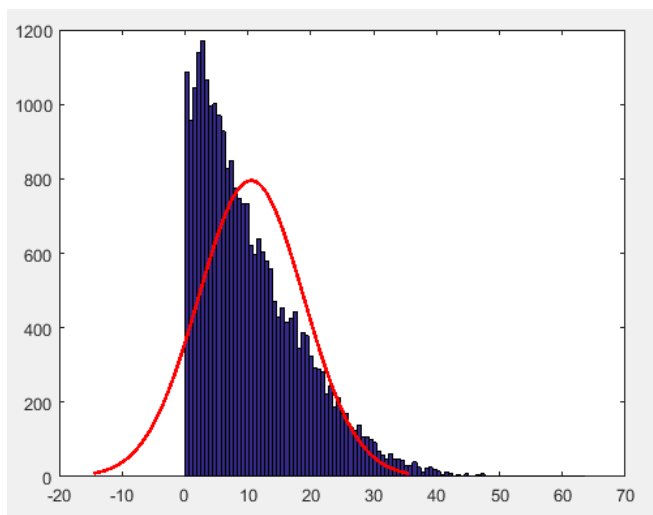
**Gambar 5.3.52** Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre jazz



**Gambar 5.3.53** Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre pop



**Gambar 5.3.54** Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre reggae



**Gambar 5.3.55** Histogram nilai fitur short time energy pada berkas mp3 musik dengan genre rock

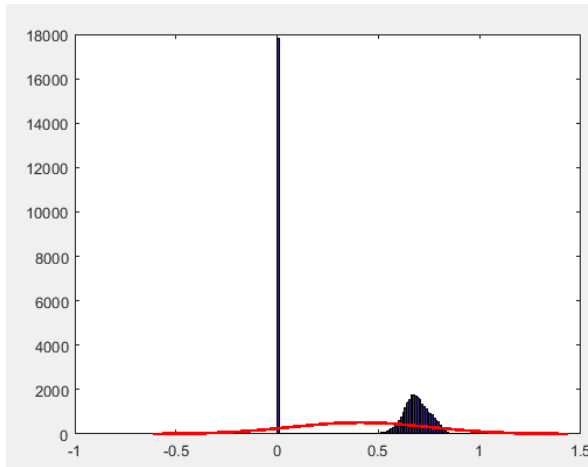
**Tabel 5.3.5** Hasil perhitungan nilai statistik data nilai fitur short time energy pada frame-frame yang terbentuk

| Jenis berkas | Mean  | Standard Deviasi | Skewness | Kurtosis |
|--------------|-------|------------------|----------|----------|
| Metal        | 22.79 | 15.49            | 1.36     | 5.62     |
| HipHop       | 11.46 | 17.49            | 2.08     | 6.87     |
| Country      | 10.84 | 6.53             | 1.16     | 5.88     |
| Rock         | 10.52 | 8.35             | 1.07     | 4.01     |
| Pop          | 10.08 | 10.48            | 1.62     | 5.97     |
| Reggae       | 9.98  | 12.43            | 3.11     | 17.18    |
| Disco        | 9.81  | 9.12             | 1.96     | 8.08     |
| Jazz         | 5.78  | 4.27             | 1.62     | 7.41     |
| Blues        | 2.18  | 2.25             | 2.09     | 9.45     |
| Audio book   | 0.33  | 0.57             | 3.37     | 23.73    |
| Classical    | 0.01  | 0.01             | 3.56     | 25.29    |

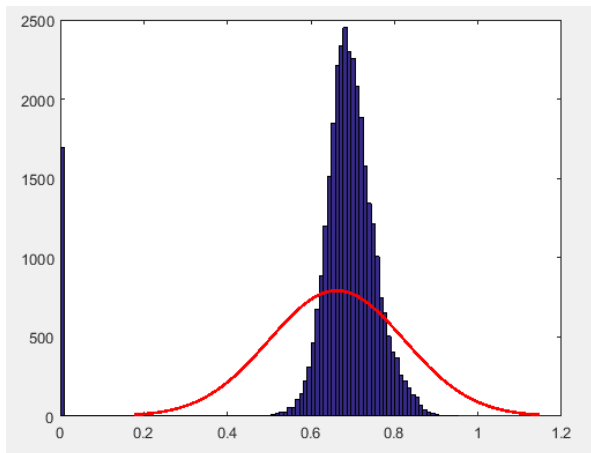
Data hasil penghitungan nilai statistik fitur spectral rolloff pada tiap berkas mp3 dengan genre yang dibandingkan ditunjukkan pada Tabel 5.3.5. Dari data Tabel 5.3.5 dapat disimpulkan bahwa semakin keras atau ramai suatu lagu maka nilai *mean* data nilai fitur short time energy akan semakin besar.

### 5.3.6 Uji Coba Ekstraksi Fitur Spectral Flatness

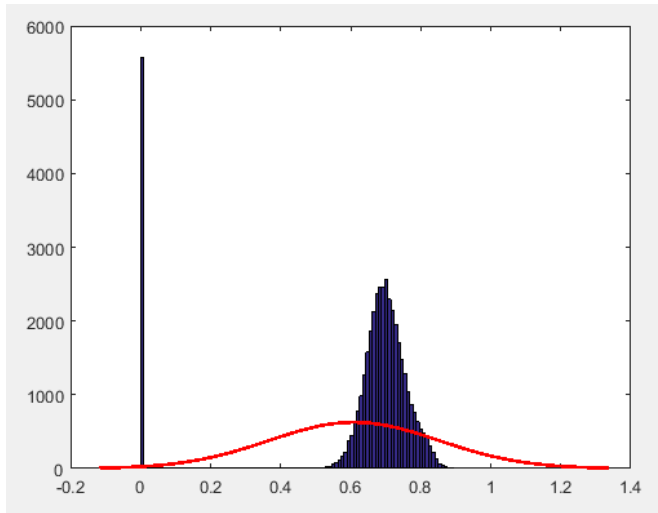
Pada subbab ini, dijelaskan mengenai hasil uji coba perbandingan bentuk histogram data nilai fitur spectral flatness di tiap frame-frame yang terbentuk dengan jenis berkas musik yang berbeda genre atau karakteristik. Perbandingan histogram antar genre pada fitur spectral flatness ditunjukkan pada Gambar 5.3.56 hingga Gambar 5.3.66.



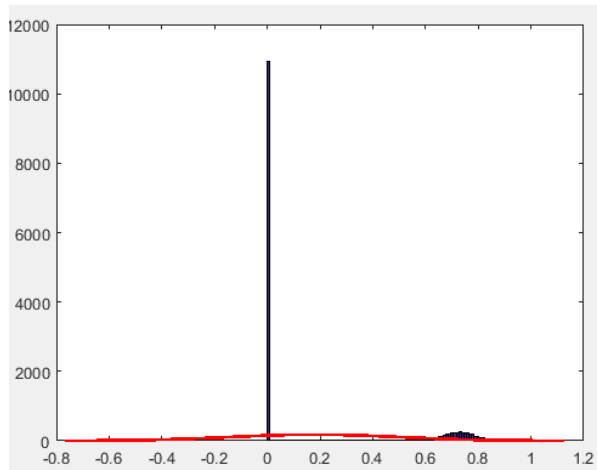
**Gambar 5.3.56** Histogram nilai fitur short time energy pada berkas mp3 buku audio



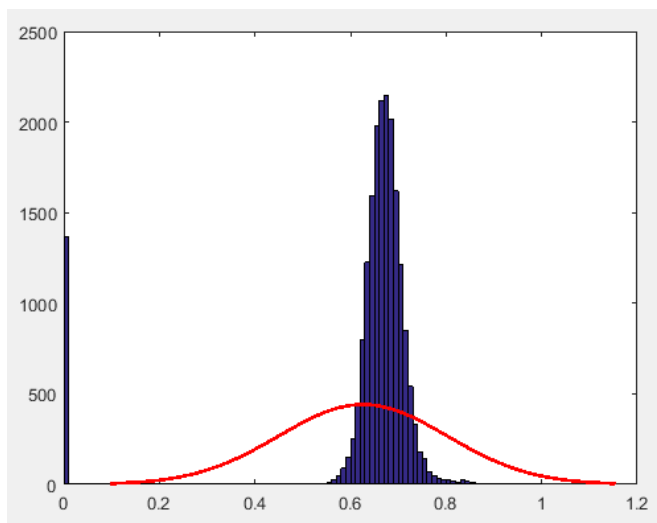
**Gambar 5.3.57** Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre metal



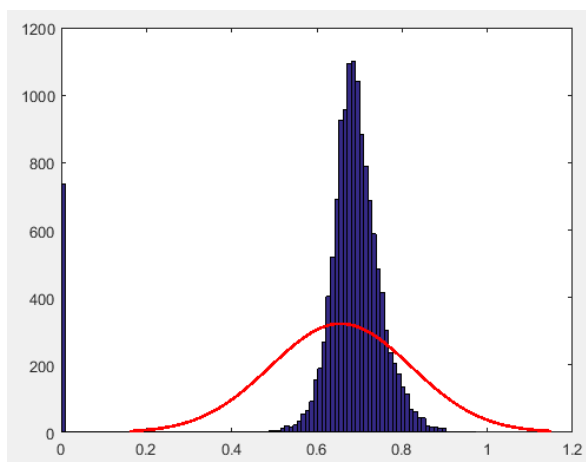
**Gambar 5.3.58** Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre blues



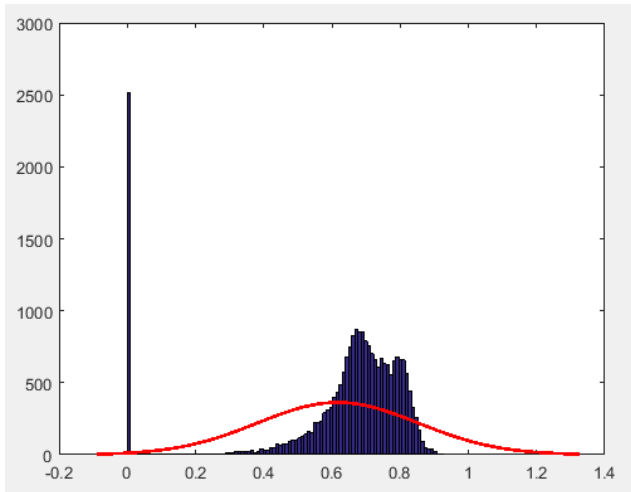
**Gambar 5.3.59** Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre classical



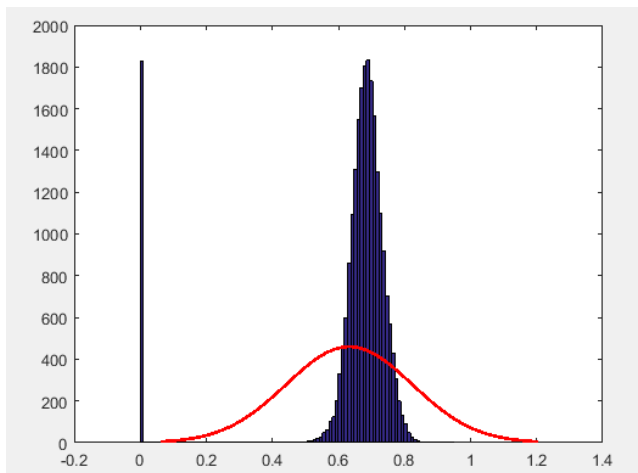
**Gambar 5.3.60** Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre country



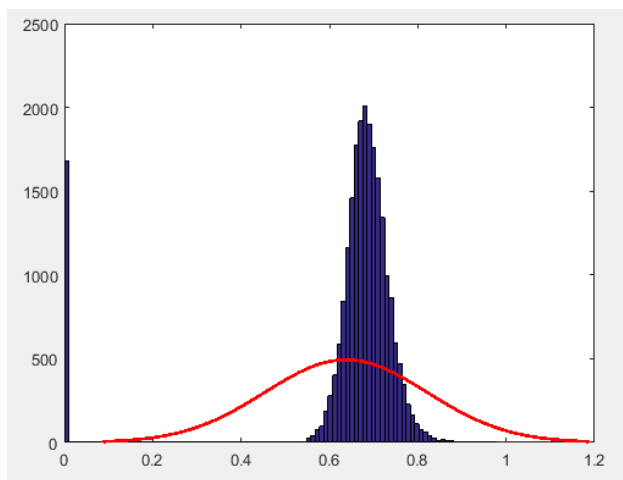
**Gambar 5.3.61** Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre disco



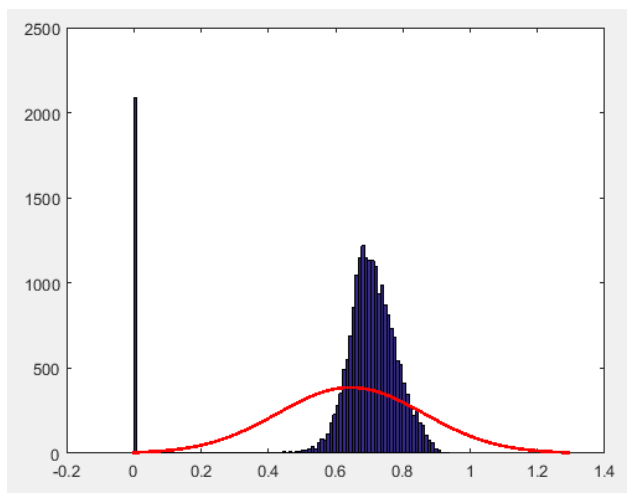
**Gambar 5.3.62** Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre hip hop



**Gambar 5.3.63** Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre jazz

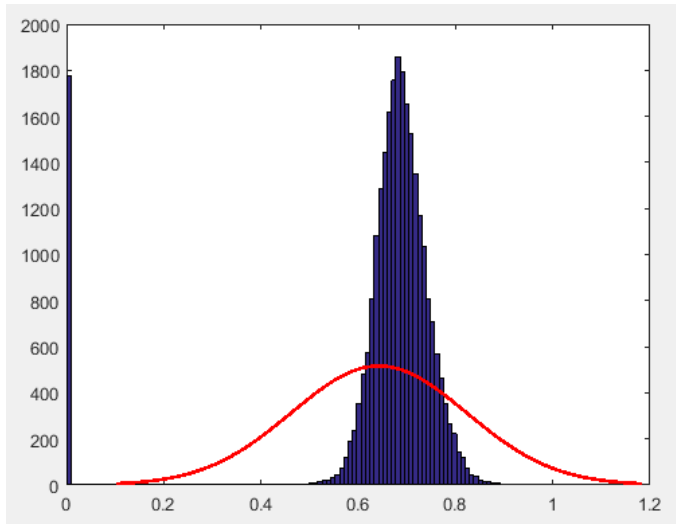


**Gambar 5.3.64** Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre pop



**Gambar 5.3.65** Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre reggae





**Gambar 5.3.66** Histogram nilai fitur spectral flatness pada berkas mp3 musik dengan genre rock

## 5.4 Skenario dan Evaluasi Pengujian

Pada subbab ini dijelaskan mengenai skenario uji coba yang akan dilakukan. Uji coba ini dilakukan untuk menguji apakah hasil ekstraksi fitur musik dengan metode yang telah dilakukan dapat digunakan untuk mengelompokkan file musik berdasarkan genre. Terdapat beberapa skenario uji coba yang akan dilakukan, yakni sebagai berikut:

1. Perbandingan nilai akurasi hasil klasifikasi dengan berdasarkan pada panjang *step* yang digunakan saat proses ekstraksi fitur.
2. Perbandingan hasil akurasi berdasarkan kombinasi data genre musik yang diolah.
3. Perbandingan hasil akurasi berdasarkan metode klasifikasi yang digunakan.

### 5.4.1 Uji Coba dan Evaluasi Skenario 1

Pada skenario ini dilakukan uji coba dengan membandingkan nilai *step* yang digunakan ketika melakukan proses ekstraksi fitur. Uji coba ini dilakukan dengan tujuan untuk mengevaluasi nilai akurasi atau nilai kebenaran dari perbedaan nilai *step* yang digunakan, yang mana nilai *step* ini mempengaruhi nilai fitur tiap file musik pada saat proses ekstraksi fitur. Dengan kata lain, hasil dari uji coba skenario ini adalah berupa nilai *step* terbaik yang kemudian akan digunakan sebagai acuan untuk proses uji coba selanjutnya.

Istilah *step* pada Tugas Akhir ini menunjukkan jarak data antara titik awal *frame* satu ke titik awal *frame* selanjutnya dalam satuan presentase panjang *frame*. *Step* yang nilainya kurang dari panjang *frame* menyebabkan data yang diolah menjadi tumpang tindih.

Uji coba perbandingan hasil akurasi berdasarkan nilai *step* pada proses ekstraksi fitur musik dilakukan dengan menggunakan 700 file musik sebagai data *training*, dan 300 file musik sebagai data *testing* yang diambil dari website <http://freemusicarchive.org>. Uji coba dilakukan dengan membandingkan dua nilai *step*, yakni 0.5 dan 1. Uji coba pada skenario pertama ini menggunakan data musik dengan genre classical, country, hip hop, jazz, metal, pop, dan rock. Dan parameter jumlah fitur yang digunakan pada saat proses klasifikasi dengan *Random Forest* sejumlah 4 fitur.

Kemudian dilakukan proses *training* menggunakan alat bantu Weka dengan metode *Random Forest*. Data *training* dan data *testing* menggunakan data yang telah disiapkan di awal. Hasil *confussion matrix* yang didapat setelah proses klasifikasi *Random Forest* dengan menggunakan nilai *step* 0.5 dan 1 saat ekstraksi fitur ditunjukkan pada tabel 5.4.1 dan tabel 5.4.2. Pada *confusion matrix* tersebut, ditunjukkan jumlah file musik yang tergolong pada genre yang diprediksikan dan dibandingkan dengan genre yang benar.

Perbandingan hasil akurasi dari setiap nilai *step* ditunjukkan pada tabel 5.4.3. Pada *step* dengan nilai 1, memiliki

nilai akurasi kebenaran lebih besar jika dibandingkan dengan nilai *step* 0.5, yakni sebesar 80%. Dengan kata lain, hasil akurasi terbaik dicapai dengan menggunakan nilai *step* pada saat ekstraksi fitur sebesar 1.

**Tabel 5.4.1** *Confusion matrix* hasil klasifikasi *Random Forest* dengan nilai *step* saat ekstraksi fitur sebesar 0.5

| Prediksi  | Kenyataan |         |        |      |       |     |      |
|-----------|-----------|---------|--------|------|-------|-----|------|
|           | classical | country | hiphop | jazz | metal | pop | rock |
| classical | 2         | 1       | 0      | 0    | 0     | 0   | 4    |
| country   | 25        | 29      | 0      | 0    | 0     | 0   | 7    |
| hiphop    | 0         | 0       | 30     | 0    | 0     | 0   | 0    |
| jazz      | 0         | 0       | 0      | 24   | 0     | 0   | 2    |
| metal     | 0         | 0       | 0      | 6    | 26    | 0   | 4    |
| pop       | 0         | 0       | 0      | 0    | 4     | 30  | 2    |
| rock      | 0         | 0       | 0      | 0    | 0     | 0   | 11   |

**Tabel 5.4.2** *Confusion matrix* untuk hasil klasifikasi *Random Forest* dengan nilai *step* saat ekstraksi fitur sebesar 1

| Prediksi  | Kenyataan |         |        |      |       |     |      |
|-----------|-----------|---------|--------|------|-------|-----|------|
|           | classical | country | hiphop | jazz | metal | pop | rock |
| classical | 3         | 4       | 0      | 0    | 0     | 0   | 0    |
| country   | 19        | 25      | 0      | 0    | 0     | 0   | 0    |
| hiphop    | 1         | 0       | 30     | 0    | 0     | 0   | 0    |
| jazz      | 7         | 1       | 0      | 24   | 0     | 0   | 0    |
| metal     | 0         | 0       | 0      | 6    | 28    | 0   | 0    |
| pop       | 0         | 0       | 0      | 0    | 2     | 30  | 2    |
| rock      | 0         | 0       | 0      | 0    | 0     | 0   | 28   |

**Tabel 5.4.3** Nilai akurasi hasil proses klasifikasi *Random Forest* uji coba skenario 1

| Nilai <i>Step</i> | 0.5       | 1   |
|-------------------|-----------|-----|
| Nilai Akurasi     | 72.8571 % | 80% |

#### 5.4.2 Uji Coba dan Evaluasi Skenario 2

Pada skenario ini dilakukan uji coba dengan membandingkan beberapa kombinasi genre musik yang dilibatkan saat proses klasifikasi. Tujuan dilakukannya uji coba adalah untuk mengevaluasi nilai akurasi atau nilai kebenaran berdasarkan kombinasi data musik berdasarkan genre yang dilibatkan pada data *training* dan data *testing*, sehingga dapat diketahui kombinasi genre apa saja yang paling cocok untuk direpresentasikan dengan metode ekstraksi fitur pada Tugas Akhir ini. Uji coba skenario 3 ini menggunakan kombinasi dari 1000 buah berkas musik dengan 10 genre yaitu genre musik blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, dan rock yang diunduh dari website <http://freemusicarchive.org>. Uji coba perbandingan dilakukan dengan tujuan untuk mengevaluasi kombinasi genre apa saja yang paling sesuai untuk fitur yang telah diekstraksi. Dengan kata lain hasil dari uji coba skenario adalah berupa kombinasi genre terbaik yang kemudian akan digunakan sebagai acuan untuk proses uji coba selanjutnya.

Uji coba perbandingan hasil akurasi berdasarkan kombinasi data genre yang digunakan pada data *training* dan data *testing* dilakukan dengan menggunakan 700 berkas musik sebagai data *training*, dan 300 file musik sebagai data *testing* yang diambil dari website <http://freemusicarchive.org>.

Dengan didasarkan hasil uji coba pada skenario pertama, maka besar *step* yang digunakan adalah 1. Uji coba pada skenario dua dilakukan dengan membandingkan hasil akurasi terhadap sekumpulan data musik dengan 2 kombinasi genre. Kombinasi genre tersebut adalah sebagai berikut :

1. Classical, country, hiphop, jazz, metal, pop, rock.  
(Kombinasi Genre 1)
2. Blues, country, disco, jazz, metal, pop, reggae.  
(Kombinasi Genre 2)

Setelah proses *training* dilakukan menggunakan alat bantu Weka dengan metode *Random Forest*. Data *training* dan data *testing* menggunakan data yang telah disiapkan di awal. Hasil *confussion matrix* yang didapat setelah proses klasifikasi *Random Forest* pada uji coba skenario kedua ditunjukkan pada Tabel 5.4.4 hingga Tabel 5.4.5. Pada *confusion matrix* tersebut, ditunjukkan jumlah file musik yang tergolong pada genre yang diprediksikan oleh hasil klasifikasi *Random Forest* dan dibandingkan dengan genre yang sebenarnya.

**Tabel 5.4.4** *Confusion matrix* untuk hasil uji coba skenario ketiga dengan menggunakan data Kombinasi Genre 1

| Prediksi  | Kenyataan |         |        |      |       |     |      |
|-----------|-----------|---------|--------|------|-------|-----|------|
|           | classical | country | hiphop | jazz | metal | pop | rock |
| classical | 7         | 0       | 0      | 0    | 0     | 0   | 0    |
| country   | 16        | 28      | 0      | 0    | 0     | 0   | 0    |
| hiphop    | 0         | 1       | 30     | 0    | 0     | 0   | 0    |
| jazz      | 7         | 1       | 0      | 23   | 0     | 0   | 0    |
| metal     | 0         | 0       | 0      | 6    | 28    | 0   | 0    |
| pop       | 0         | 0       | 0      | 1    | 2     | 30  | 2    |
| rock      | 0         | 0       | 0      | 0    | 0     | 0   | 28   |

**Tabel 5.4.5** *Confusion matrix* untuk hasil uji coba skenario ketiga dengan menggunakan data Kombinasi Genre 2 (Bagian pertama)

| Prediksi | Kenyataan |         |       |      |       |     |        |
|----------|-----------|---------|-------|------|-------|-----|--------|
|          | blues     | country | disco | jazz | metal | pop | reggae |
| blues    | 11        | 0       | 0     | 0    | 0     | 0   | 0      |
| country  | 2         | 29      | 10    | 0    | 0     | 0   | 0      |
| disco    | 0         | 1       | 18    | 4    | 0     | 0   | 11     |

**Tabel 5.4.6** *Confusion matrix* untuk hasil uji coba skenario ketiga dengan menggunakan data Kombinasi Genre 2 (Bagiam kedua)

|        |    |   |   |    |    |    |    |
|--------|----|---|---|----|----|----|----|
| jazz   | 17 | 0 | 2 | 15 | 0  | 0  | 0  |
| metal  | 0  | 0 | 0 | 6  | 23 | 1  | 0  |
| pop    | 0  | 0 | 0 | 0  | 2  | 25 | 2  |
| reggae | 0  | 0 | 0 | 5  | 4  | 4  | 17 |

**Tabel 5.4.7** Nilai akurasi hasil proses klasifikasi  
*Random Forest* uji coba skenario 2

| Kombinasi Genre   | Nilai Akurasi |
|-------------------|---------------|
| Kombinasi Genre 1 | 80 %          |
| Kombinasi Genre 2 | 65.7143 %     |

Perbandingan hasil akurasi dari setiap kombinasi data berdasarkan genre ditunjukkan pada tabel 5.4.7. Pada Kombinasi Genre 1, ditunjukkan bahwa memiliki nilai akurasi kebenaran lebih besar jika dibandingkan dengan kombinasi genre lainnya, yakni sebesar 80 %. Dengan kata lain, hasil akurasi terbaik dicapai dengan menggunakan kombinasi data genre 1, yakni genre classical, country, hiphop, jazz, metal, pop, rock.

### 5.4.3 Uji Coba dan Evaluasi Skenario 3

Pada skenario ini dilakukan uji coba dengan membandingkan nilai akurasi hasil ekstraksi fitur yang dilakukan dengan beberapa metode klasifikasi. Tujuan dilakukannya uji coba adalah untuk membandingkan nilai akurasi atau nilai kebenaran hasil ekstraksi fitur dengan metode klasifikasi *naive bayes*, *random forest*, *multi layer perceptron*, *support vector machine*, dan J48. Uji coba skenario 3 ini menggunakan kombinasi dari 1000 buah berkas musik dengan 7 genre yaitu genre musik classical, country, hiphop, jazz, metal, pop, rock yang diunduh dari website <http://freemusicarchive.org>. Uji coba perbandingan

dilakukan dengan tujuan untuk metode klasifikasi mana yang paling sesuai untuk fitur yang telah diekstraksi. Dengan kata lain hasil dari uji coba skenario adalah berupa nilai akurasi dari hasil klasifikasi dengan metode *naive bayes*, *random forest*, *multi layer perceptron*, *support vector machine*, dan J48.

**Tabel 5.4.8** Perbandingan nilai akurasi berdasarkan metode klasifikasi yang digunakan pada skenario 3

| No. | Metode Klasifikasi     | Nilai Akurasi (%) |
|-----|------------------------|-------------------|
| 1   | Naïve Bayes            | 80.4762           |
| 2   | Random Forest          | 80                |
| 3   | Multi Layer Perceptron | 77.619            |
| 4   | Support Vector Machine | 69.5238           |
| 5   | J48                    | 62.8571           |

Perbandingan hasil akurasi dari setiap kombinasi data berdasarkan genre ditunjukkan pada tabel 5.4.8. Dari tabel tersebut ditunjukkan bahwa metode klasifikasi Naïve Bayes menghasilkan nilai akurasi tertinggi sebesar 80.4672% dan dilanjutkan dengan metode random forest dengan nilai akurasi 80%.

*[Halaman ini sengaja dikosongkan]*



## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

#### **6.1. Kesimpulan**

Dari hasil uji coba yang telah dilakukan terhadap pembuatan program ekstraksi fitur musik berdasarkan genre dapat diambil kesimpulan sebagai berikut:

1. Fitur yang dihasilkan saat proses ekstraksi fitur musik pada Tugas Akhir ini mampu merepresentasikan musik berdasarkan genre dengan baik, terbukti dengan nilai akurasi dengan prosentase 80.4762 % ketika diklasifikasi dengan metode naïve bayes.
2. Dalam kasus pengelompokkan berkas musik berdasarkan genre, nilai  $step = 1$  merupakan yang paling bagus jika dibandingkan dengan nilai step yang lain ketika digunakan saat proses ekstraksi fitur untuk menghasilkan akurasi yang tinggi.

#### **6.2. Saran**

Saran yang diberikan untuk pengembangan dari implementasi ekstraksi fitur musik berdasarkan genre pada Tugas Akhir ini antara lain:

1. Oleh Karena nilai akurasi dari hasil proses klasifikasi sebesar 80.4762 %, maka perlu dilakukan perbaikan terhadap kualitas fitur yang sudah ada ataupun menambahkan fitur lain yang dapat lebih merepresentasikan genre pada suatu file musik.

2. Perangkat lunak yang dikerjakan pada Tugas Akhir ini dapat dikembangkan menjadi berbasis web maupun *mobile* agar dapat dimanfaatkan secara langsung oleh masyarakat umum.

## DAFTAR PUSTAKA

- [1] D. G. J. L. Babu Kaji Baniya, "Automatic Music Genre Classification Using Timbral Texture and Rhythmic Content Features," *ICACT Transactions on Advanced Communications Technology (TACT)*, vol. 3, no. 3, 2014.
- [2] M. Rouse, "WhatIs.com," [Online]. Available: <http://whatis.techtarget.com/definition/MP3-MPEG-1-Audio-Layer-3>. [Accessed 12 1 2017].
- [3] "The Free Music Archive," [Online]. Available: <http://freemusicarchive.org/>. [Accessed 5 12 2016].
- [4] George Tzanetakis, Georg Essl, Pery Cook, "Automatic Musical Genre Classification Of Audio Signals," *IEEE Transactions on Speech and Audio Processing*, pp. 293-302, 2001.
- [5] "Fast Fourier Transform - MATLAB & Simulink," [Online]. Available: <https://www.mathworks.com/help/matlab/ref/fft.html>. [Accessed 14 12 2016].
- [6] "Hamming Window - MATLAB hamming," [Online]. Available: <https://www.mathworks.com/help/signal/ref/hamming.html>. [Accessed 14 12 2016].
- [7] L. Breiman and A. Cutler, "Statistics at UC Berkeley, Department of Statistics," [Online]. Available: [http://www.stat.berkeley.edu/~breiman/RandomForests/cc\\_home.htm](http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm). [Accessed 1 December 2016].
- [8] L. BREIMAN, "Random Forests," in *Machine Learning*, Kluwer Academic Publishers, 2001, p. 5–32.

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Ramadhan Rosihadi Perdana, lahir di Surabaya, pada tanggal 12 Maret 1994. Penulis menempuh pendidikan mulai dari SDN 1 Ketandan (1999-2005), SMP Negeri 1 Dolopo (2005-2008), SMA Negeri 1 Geger (2008-2011) dan S1 Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember, Surabaya (2012-2016).

Selama masa kuliah, penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer (HMTTC). Diantaranya adalah menjadi staf Departemen Pengembangan Profesi HMTTC ITS 2013-2014. Penulis juga aktif dalam kegiatan kepanitiaan Schematics 2014, diantaranya penulis pernah menjadi kepala biro *sponsorship*.

Selama kuliah di Teknik Informatika ITS, penulis mengambil bidang minat Kecerdasan Citra dan Visual (KCV). Penulis dapat dihubungi melalui alamat surel **ramadhanrperdana@gmail.com**